DETERMINATION OF OPTIMAL ROUTES AND VEHICLE DISPATCHING STRATEGIES FOR A DISTRIBUTION NETWORK

By
JAGJIWAN LAL MODI

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

SEPTEMBER, 1974

DETERMINATION OF OPTIMAL ROUTES AND VEHICLE DISPATCHING STRATEGIES FOR A DISTRIBUTION NETWORK

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

By
JAGJIWAN LAL MODI

to the

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

SEPTEMBER, 1974

ME-1974-M-MOD-DET

CERTIFICATE

Certified that this work on 'Determination of Optimal Routes and Vehicle Dispatching Strategies for a Distribution Network' by Jagjiwan Lal Modi has been carried out under my supervision and that this has not been submitted elsewhere for a degree.

Kanpur
September 1974

Dr. J.L. Batra
Assistant Professor
Department of Mechanical Engineering
Indian Institute of Technology, Kanpur

ACKNOWLEDGEMENT

I wish to express my indebtness to Dr. J.L. Batra who has been a constant inspiration and a guide in the completion of this work.

I am thankful to the management, Kanpur Sahakari Milk Board, especially Mr. Bias, Assistant Sales Manager, for his patient help in data collection.

Thanks are due to the Computer Centre staff who have always been there with help at times of depression.

Last, but not the least, I am thankful to Mr. Kumar of Tata Consultacy Services, Bombay for taking pains to type the manuscript patiently and neatly.

Kanpur
September 1974

- Jagjiwan Lal Modi

Dedicated to

My Parents and Gudiya

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SYNOPSIS

Jagjiwan Lal Modi
M.Tech. (Mech.)
Indian Institute of Technology, Kanpur
September 1974

"DETERMINATION OF OPTIMAL ROUTES AND VEHICLE DISPATCHING STRATEGIES FOR DISTRIBUTION NETWORK"

In the present work, an heuristic approach has been developed for designing the optimal routes through customers in a distribution network, for selecting the optimal set of vehicles and for their allocation to the routes designed so as to satisfy customers' demands at the minimum distribution cost. The proposed model can handle the allocation of equal as well as multiple capacity vehicles. This model also takes into account the time constraint - that has not been considered heretofore - for ensuring the distribution of commodity to customers within prescribed time.

Method of solution starts with constructing initial routes by Multiple Travelling Salesman Algorithm. Initial routes are adjusted for the capacity and time constraint, keeping the distribution cost at the minimum possible level. Finally, the refining heuristic are used to improve the routes in order to reduce the distribution cost. A computer package has been developed for the proposed methodology.

Nine test problems have been solved using the proposed methodology. Same set of problems have been solved using Clarke and Wright's method. A comparison of the results

indicate the superiority of the proposed methodology over Clarke & Wright's method in terms of total distribution cost. Further, the proposed model has been tested on a case study based on the distribution system of Kanpur Sahakari Milk Board. It is observed that if the routes and dispatching strategies obtained by the proposed methodology are used, it results in a saving over the currently used policies of Kanpur Sahakari Milk Board. Savings have been observed to vary from 20% to 32% depending upon the time period under consideration.

CHAPTER I

INTRODUCTION

The primary objective of any business organization is to make money. Managers are always on the look out for finding ways and means to reduce the costs of production and distribtution so that product reaches the customer at the lowest possible price. If a business is to succeed it must exert no less effort towards the efficient direction and control of distribution costs than towards the production costs. Therefore, most business executives are trying to find ways and means to reduce distribution costs. In many concerns distribution cost constitutes the major cost and in most industries it is a factor of major proportion.

An efficient and economic distribution system is particularly very important for those industries in which the distribution expense represents a major share of the price paid by the ultimate consumer. When there is a buyer's market in any industry, the low-cost distributor has an important edge on competition. Inefficient distribution system leads to reductions in profit and/or dissatisfaction of customers.

Distribution embraces all the activities in an enterprise that are required to move finished goods from the ends of production lines to the points of ultimate sake or use. In an industry which uses vehicles for the distribution of its goods, an attempt for getting an efficient distribution system creats the vehicle-dispatching problem i.e., the problem of designing optimal routes through customers in distribution network, finding an optimal size of fleet and its allocation to the routes

designed so as to satisfy customers' demands at the minimum delivery cost. Present work attempts to solve such a problem.

Literature on vehicle dispatching problems can be traced back to 1960 when Dantzig & Ramzer (8) attempted the problem as a linear program. Later on, the problem was also formulated in Integer programming terms (1) but large number of variables and constraints precluded the solution of such models. This led to the development of heuristic approaches which could successfully handle real life distribution problems which happened to be large in size. It is to be pointed out that most of the currently available approaches give near optimal solution to the problem. Therefore, better algorithms need to be developed for the efficient generations of optimal solutions. Further, a thorough survey of the literature indicated that most of the research workers have concentrated on models with assumptions of constant capacity vehicles and single product delivery. No extensive work has been reported for the problems incorporating traffic constraints in distribution network and time constraint ensuring the delivery of commodities to customer points in prescribed time. On the other hand, if the model is to be of any practical significance it should also incorporate the cases of multiple capacity vehicles, multi-product distribution, traffic and time constraint etc. The literature review reveals that, but for a brief mention, no serious attempts have been made by the researchers to incorporate the above listed features into their models. In the present work, some of these features have been incorporated. An approach, which is heuristic in character, has been developed for solving vehicle dispatching problems with vehicles having same capacities as well as different capacities and incorporating the constraints for (1) customers' demand satisfaction (2) maximum load on a route depending upon the capacity of vehicle on that route and

(3) route and vehicle time for distribution of commodity to customers within prescribed time. The proposed methodology has been validated using a case study on the distribution system of Kanpur Sahakari Milk Board. Further, the proposed method has been compared with that of Clarke & Wright's method (6) for nine test problems in which the time constraint has been relaxed. The practical application of the proposed model extends to all distribution systems where a fleet of delivery vehicles deliver a commodity from a supplying source to a set of customers with known location.

CHAPTER-II

LITERATURE SURVEY

2.1 A Critical Review:

The problem of producing routes for vehicles servicing a number of customers from a single depot has aroused much interest among the researchers. In general, the objective is to produce a set of routes which minimize the total cost of delivery subject to the restrictions due to the maximum load carrying capacity of the vehicles and the maximum allowable elapsed time (or distance) for any route. The cost factors generally considered are the number of trucks and the total distance travelled to satisfy the customer's demands. The approaches that are cited in literature for solving such problems are discussed in the paragraphs that ensue.

Balinski and Quandt (1) formulated the above problem as an Integer Program. They listed various possible feasible combinations depending upon the number of customers, number of permissible geographical routes and number of customers delivered together by a given vehicle and computed their costs. Problem was then to find a set of activities that satisfy all orders and minimizes total cost. They defined an activity to be a single feasible combination of customers' demands. This model is not useful for large size practical problems because of high number of variables involved.

Dantzig and Ramser (8) proposed a linear programming approach for the truck dispatching problems. The objective was to minimize the interpair distances between delivery points. Their method of solution starts from the basic idea to synthesize the solution in a number of

'Stages of aggregation' in which suboptimizations are carried out on pairs of delivery points or groups. This method tends to lay more emphasis on filling the delivery trucks to near capacity than on minimizing total distance. Therefore, it does not give optimal solution and sometimes results might be far from optimal.

Clarke and Wright (6) developed an heuristic approach to incorporate the practical constraints and formed the basis for most methods developed to date. This approach is called "Savings approach" because in this method the savings for each possible link are calculated and arranged in descending order and then each link is examined in turn by going down the list. If the link under consideration satisfies the constraints of capacity and distance, it is added, otherwise another link is examined, until no more links can be added and we are then left with the final tours. Inspite of its simplicity for use, this method does not guarantee optimality for vehicle routes. Nevertheless, the results obtained by this method become progressively worse as the constraints are made more stringent. Gaskell (15) has shown that the results obtained by Clarke & Wrights' method are often far from optimal results. This approach has been successfully used, for some cases, by Norman (28) and by National Computing Centre (29).

Christofides and Eilon (4) generalized the simple methods of solution of the "Travelling Salesman Problem" for solving the truck dispatching problems. They used two techniques namely Branch and Bound technique and "r-optimal tour" technique. Christofides and Eilon observed that the Branch and Bound technique was not practicable for large size vehicle dispatching problems because it required excessive computer time and memory. Further, they found that the computational efficiency of the Branch and Bound algorithm when applied to the vehicle scheduling problems was substantially reduced compared with its efficiency in solving an equivalent Travelling Salesman Problem. Therefore, for large

size problems, they developed 'r-optimal tour' method. They defined 'r-optimal tour' as the tour which cannot be reduced in length by replacing r of its links by r other links. Their method starts with an arbitrary random route structure and each route is examined to see whether mileage can be reduced by re-arranging the customers of that route. The method is then repeated using different initial solutions. In general, the probability that the r-optimal tour is minimal, increases with the value of r. But on other hand, the amount of computations required increases very rapidly with value of r. They claim that 3-optimal tour produced very good results. To substantiate this claim, Christofides and Eilon solved ten problems and compared the results with the results obtained by Clarke and Wrights approach. The results were found to be quite comparable with Clarke & Wright's approach.

The approaches discussed above, are very restrictive in nature because they cannot handle situations when a customer may be served by one of the several depots. Recently, Wren and Holliday (34) presented an algorithm which allows routes from several depots to be constructed simultaneously subject to restrictions on number of vehicles at individual depots and constraints on load and distance. Their approach is first to construct initial feasible routes and then to apply a number of different refining heuristics to the initial routes. The quality of final routes depends very much upon the initial routes constructed. For constructing initial routes, the customers are presented to the algorithm in a predetermined order which affects the quality of initial routes. Refining heuristics further improve the initial routes (improvement being measured in terms of reduction of distance or number of vehicles). Theoretically, the programme might be allowed to continue until none of heuristicscould produce an improvement. However, they found that after a spectacular initial leap, the rate of improvement often slowed down considerably. Therefore, they suggested that the program should be terminated after a specified amount of computer time.

2.2 Summary

The review of literature has revealed a number of approaches developed for tackling dispatching problem. The limitations and the relative performances of different models in terms of results obtained and computational efficiencies have also been discussed. However, but for a brief mention, none of these approaches handle the problem of route selection and vehicle dispatching when the delivery vehicles are of different capacities, multiple products are to be delivered to the customers in a pre-specified duration of time.

CHAPTER-III

MATHEMATICAL FORMULATION

3.1 Problem Description

Broadly speaking, the problem can be viewed as designing the optimal delivery routes and the determination of the optimal set of vehicles to be plied on these routes out of the available vehicles of known capacities. Delivery vehicles distribute the commodity from the depot to a set of customers each with a known location and anticipated demand.

Before the mathematical formulation is presented, author feels it in order to discuss briefly the objective function, the various constraints and the assumptions made for the development of the model.

3.1.1 Objective Function

The objective is to minimize total cost of distribution to satisfy a known demand. The various distribution costs are :

1) Cost of Transportation: This is the cost of fuel consumed by vehicles in transporting the commodity to customers. Obviously the transportation cost for a delivery vehicle is approximately a linear function of distance travelled by that vehicle.

2) Labour Maintenance and Depreciation Costs: Labour cost includes the wages for drivers, helpers and route clerks involved in distribution operations. This cost is a linear function of number of vehicles in operation, because each vehicle is accompanied by driver, route clerk and helpers. Maintenance cost is the cost for maintaining the vehicles for keeping them ready for the operations. It depends upon the condition

of vehicle, frequency of accidents etc. Depreciation cost refers to the fall in value of the vehicle with respect to time.

3.1.2 Constraints:

1) Demand Constraint : Routes designed should be such that anticipated demand at each customer point is fully satisfied. In case, the management decides to fix up some quota to be delivered to the customers, demand is considered as to be equal to that quota.

2) Capacity Constraint: Vehicle routes should be designed such that the total amount to be delivered by any vehicle, on the route allocated to it, should not be more than its capacity.

3) Time Constraint: This constraint is of much importance when commodity to be delivered to customer is perishable one. This constraint is incorporated to restrict the delivery to each customer within permissible time limits.

i) Route time constraint: The upper limit for the time elapsed on a route is kept to make sure of deliveries to all customers during the specified time. Route time includes total travelled time on the route and the total stop-off-time at customers' points on that route. This constraint restricts the length of the route and the number of customers to be served on a route.

ii) Vehicle time constraint: This constraint comes into picture when a vehicle is assigned to more than one route In this case, vehicle has to return to depot, after delivering one route, to get loaded for serving next route assigned to it. So loading and unloading of vehicl

at depot is also involved. The total time elapsed from the moment the vehicle is loaded for serving first route assigned to it, to the time this vehicle is unloaded at depot after serving the last route assigned to it, is the vehicle time because for that much time vehicle was in operation continuously. This constraint restricts the number of routes which can be served by the same vehicle. Significance of this constraint is when delivery vehicles are being operated on a shift basis.

At this stage, it is opportune to explain some of the terms used above.

a) Capacity of Vehicle: Capacity of any vehicle is the maximum permissible load (in terms of weight or volume) which can be carried safely by that vehicle. Capacity of vehicle is also affected by the condition of vehicle.

b) Stop-off-time at each customer point : Stop-off-time at a customer point includes

i) time for unloading the items to be delivered to that customer.

ii) time for loading the items returned from that customer.

iii) time for preparing bill for money transaction etc.

In General, stop off time at each customer point is different, depending upon the amount of loading and unloading involved.

c) Stop-off-time at depot: It includes the time for loading and unloading of the vehicle at depot.

d) Maximum permissible route time: Route time is the elapsed time for which vehicle is on that route, It includes

i) travelling time on the route

ii) stop off-time for all the customers on the route.

Maximum limit for route time is put for making the deliveries to all customers in a specified time.

e) Maximum permissible vehicle time: It is the time for which the vehicles are available for the distribution. It includes

i) loading and unloading time for the vehicle at depot.

ii) vehicle idle time.

iii) route time for the route(s) assigned to that vehicle.

f) Average speed of vehicles Speed of vehicle depends upon its size and condition. In addition to it, average speed of the vehicle depends on the traffic intensity of the route assigned to the vehicle.

3.1.3 Basic Assumptions:

1) All delivery vehicles would return to the depot after distributing the commodity to customers. This is usual practice of most of the companies engaged in distribution.

2) Only one route passes through a customer and full delivery to that customer is made by the vehicle assigned to that route passing through that customer. This assumption is valid for only those customers whose demand is less or equal to the capacity of vehicle. In fact, most of the distributing agencies follow this strategy.

3) All vehicles are in same condition. This assumption implies that maximum allowable load to be carried by vehicles of the same capacity is same.

4) Average permissible speed on the routes is same for a vehicle throughout the distribution network.

5) Supply from source i.e. depot is sufficient to fulfill demands of all the customers.

6) Labour and maintenance cost is same for every vehicle and therefore, the total labour and maintenance cost is considered as a linear function of the number of vehicles.

7) There is no vehicle idle time at the depot due to delay in loading and unloading of vehicle i.e., vehicle is ready for loading or unloading as soon as it reaches the depot. Similary assumption has been made that no vehicle idle time occurs on the route allocated to it.

Assumptions (3) and (4) follow that average speed of vehicles is no more a function of condition of vehicle or traffic intensity on routes. It is only a function of type of vehicle and its size.

A mathematical formulation of the above stated problem has been presented in the following section.

3.2 Mathematical Formulation

The notations and terminology used for the development of mathematical model is presented below:

3.2.1 Notations and terminology

NC - Number of customers to be served.

N - Total number of nodes in distribution network.

i	Index for the preceeding node $i = 1,2, \ldots, N$
j	Index for the suceeding node, $j = 1,2, \ldots, N$
G	Number of routes
g	Index for the routes, $g = 1,2, \ldots, G$
λg	gth route
NV	Number of vehicles employed for the distribution
d_{ij}	Shortest distance between ith & jth node
r	Index for the vehicles, $r = 1,2, \ldots, NV$.
C_r	Capacity of rth vehicle
D_j	Demand at jth node, $j = 2,3, \ldots, N$
	$D_1=0$ for $j = 1$, i.e., for the depot
y_{ij}	$= 1$ if ith node is connected to jth node in network
	$= 0$ otherwise
I_{λ_g}	Set of delivery point forming the route λg
	i.e., $j \varepsilon I_{\lambda_g}$ such that j is on route λg; for $j \neq 1$.
n_{λ_g}	Number of delivery points forming the route λg.
δ_g^r	$=$ 1 if rth verhicle is assigned to route λg
	$=$ 0 otherwise
t_{sj}	Stop off time at jth customer $j = 2,3, \ldots, N$
L_r	Loading and unloading time for rth vehicle at depot.
K_r	Cost of transportation/unit distance for rth vehicle
T_1	Maximum permissible route time.
T_2	Maximum permissible vehicle time
v_r	Average speed of rth vehicle
P	Labour and Maintenance Costs per vehicle
DC_r	Depreciation cost of rth vehicle.

3.2.2 Objective Function: Objective function is to minimize distribution costs. Total cost of distribution is determined as follows

1) Cost of transportation

Total distance travelled by vehicle on gth route,

$$TD_g = [\sum_{j\epsilon I_{\lambda g}} d_{1j}y_{1j} + \sum_{i,j\epsilon I_{\lambda g}} d_{ij}y_{ij} + \sum_{j\epsilon I_{\lambda g}} d_{j1}y_{j1}]$$

..... (1)

Total cost of transportation for gth route =

$$TC_g = \sum_{r=1}^{NV} \delta_g^r (TD_g . K_r)$$

Therefore total cost of transportation for all the routes

$$TC_1 = \sum_{g=1}^{G} \sum_{r=1}^{NV} \delta_g^r (TD_g . K_r) \qquad \text{...... (2)}$$

2) Labour and Maintenance Cost: Since the labour and maintenance cost is assumed to be the same for all the vehicles, the total labour and maintenance cost can be expressed as follows:

$$TC_2 = NV * P \qquad \text{...... (3)}$$

3) Depreciation Cost: Let DC_r be the depreciation cost for rth vehicle. Then the total depreciation cost is given by

$$TC_3 = \sum_{r=1}^{NV} DC_r \qquad \text{...... (4)}$$

• The total cost of distribution is the sum of transportation cost, labour maintenance and depreciation cost. Therefore,

$$\text{Total cost} = TC_1 + TC_2 + TC_3$$

$$= \sum_{g=1}^{G} \sum_{r=1}^{NV} \delta_g^r (TD_g * K_r) + NV * P + \sum_{r=1}^{NV} DC_r \quad \ldots (5)$$

where TD_g is distance travelled by vehicle on gth route.

3.2.3 Constraints

1) Except for depot, each one of other nodes is connected to one and only one other node. Mathematically this can be expressed as :

i) $\sum_{\substack{i=2 \\ i \neq j}}^{N} y_{ij} = 1$ for $j = 2,3,\ldots\ldots,N$ ...... (6)

ii) $\sum_{\substack{j=2 \\ j \neq i}}^{N} y_{ij} = 1$ for $i = 2,3,\ldots\ldots,N$ ...... (7)

2) Since each route is to be assigned to only one vehicle, we obtain following constraint:

$\sum_{r=1}^{NV} \delta_g^r = 1$ for $g = 1,2,\ldots\ldots,G$ ...... (8)

However, a vehicle can be assigned to many routes. Therefore,

$\sum_{g=1}^{G} \delta_g^r = K \cdot 1$ Where K is an integer value .... (9)

3) Demand Constraint: The restriction that one route should pass through a customer and full requirement of the customer is delivered at a stretch, implies that the capacity of the

vehicle assigned to the route, on which the considered customer is, should be equal to or more than demand of that customer.

i.e. $\sum_{z=1}^{NV} C_z \delta_g^z \geq D_j$ for $j = 2,3,\ldots,N$ .....(10)

where $j \in I_{\lambda g}$

Constraints (3) & (4) imply that if rth vehicle is assigned to the route, say gth, on which the customer j is, then

$$C_r \geq D_j \qquad \text{...... (11)}$$

4) Capacity Constraint: If rth vehicle is assigned to gth route then capacity of rth vehicle should be equal to or more than the sum of demands of all customers on the gth route.

i.e. $\sum_{j \in I_{\lambda g}} D_j \leq C_r$ ....(12)

In general, $\sum_{j \in I_{\lambda g}} D_j \leq \sum_{r=1}^{NV} C_r \delta_g^r$ for $g = 1, 2, \ldots G$ ......(13)

5) Time Constraint

1) Route time constraint: This constraint restricts the length of the route and number of customers to be included in that route, such that delivery to each customer is made within specified time. Route time of any route is the sum of the total time travelled by the vehicle on the route and the total stop-off-time at all the customers on that route. Let T_1 be the maximum permissible route time. Then this constraint implies that route time of any route should not exceed T_1.

Total distance travelled by vehicle on gth route,

$$TD_g = \left[\sum_{j \in I_{\lambda g}} d_{1j} y_{1j} + \sum_{i,j \in I_{\lambda g}} d_{ij} y_{ij} + \sum_{j \in I_{\lambda g}} d_{j1} y_{j1} \right] \quad \ldots (14)$$

If rth vehicle is assigned to gth route, i.e., $\delta_g^r = 1$ then route time constraint is

$$(TD_g/v_r) + \sum_{j \in I_{\lambda g}} t_{sj} \leq T_1 \quad \ldots (15)$$

In general, route time constraint is

$$\sum_{r=1}^{NV} \delta_g^r \left[(TD_g/v_r) + \sum_{j \in I_{\lambda g}} t_{sj} \right] \leq T_1 \; ; \; g=1,2\ldots.G \quad \ldots (16)$$

2) Vehicle time constraint: This constraint implies that a delivery vehicle should make all the distribution operations assigned to it within a specified time i.e., maximum permissible vehicle time, so that they are available for the next operation. Vehicle time includes the route time for the route allocated to it and loading, unloading time of vehicle at depot. Mathematically, this constraint can be expressed as,

$$\sum_{g=1}^{G} \delta_g^r \left[(TD_g/v_r) + \sum_{j \in I_{\lambda g}} t_{sj} + L_r \right] \leq T_2 \quad \text{for } r=1,2\ldots NV \quad \ldots (17)$$

Where TD_g is total distance travelled by vehicle.

6) Node-connectivity Constraint:

$y_{ij} = 1$ if ith node is connected to jth node,

for $i = 1,2,\ldots\ldots N$; $j = 1,2,\ldots\ldots N$

$= 0$ otherwise. ...... (18)

Equations (5) through (18) represent the mathematical formulation of the route selection and vehicle assignment problem under consideration. A solution methodology for the problem is presented in the next Chapter.

CHAPTER 4

METHODOLOGY

In this chapter, the methodology for solving the problem formulated in Chapter 3, is presented. The approach is heuristic in character and gives near optimal solution.

Methodology for the solution of the problem broadly consists of following phases:

(1) Determination of the shortest distances between the nodes in distribution network using shortest distance model.

(2) Selection of the optimal set of vehicles out of available vehicles.

(3) Design of optimal routes and the optimal allocation of vehicles to the various routes by Route design model.

4.1 Shortest Distance Model

Shortest path algorithm(30) has been used to find the shortest distances between the delivery points in the distribution network. The procedure is to examine simultaneously all the routes out of the starting point i and into the terminal point j as far as their adjacent connected points and to extend further the routes which have so far covered the least distance. This process is repeated step by step until a route out of i has a delivery point on it which has already occurred on a route into j or vice-versa. All such complete routes between i and j are examined and shortest possible route between i and j is found. Similarly shortest path

distance paths for each pair of nodes are determined. Computer program for the above algorithm has been written in FORTRAN IV language.

4.2 Routes Design Model

Looking back into the mathematical formulation presented in Chapter 3, it is observed that objective function, i.e., total cost of delivery depends upon (i) total distance travelled by vehicles, i.e., the sum of distances of all the routes for distribution, and (ii) the number and types of vehicles used for distribution. For a given set of vehicles, the total delivery cost depends upon the total distance travelled by vehicles in the set. So, for a particular set of vehicles, the problem consists of designing the routes for the vehicles for minimum total distance and the allocation of the vehicles to the designed routes for minimum delivery cost. As the set of vehicles is chosen from the available vehicles for distribution, there will be few feasible sets of vehicles to be considered. Problem is solved for each set of vehicles and the one which yields best results is chosen. Main steps involved are as follows:

(i) Choose a particular set of vehicles to be used for distribution.

(ii) For the above chosen set of vehicles, design the routes such that the total distance is minimum. Designing of routes is carried out in three stages:

   (a) Constructing initial routes
   (b) Adjusting initial routes for satisfying the constraints.
   (c) Improving the adjusted routes by refining heuristics.

(iii) Total cost of delivery is calculated.

(iv) Repeat the Steps (ii) and Step (iii) for various feasible set of vehicles to be used for distribution and select the best solution yielding the minimum

Next, we have to find feasible sets of vehicles which can be used for distribution.

4.2.1 Determination of feasible 'sets' of vehicles which can be used for distribution

Initially, it is assumed that each vehicle is allocated to one route only. Later on, in the refining heuristics, this assumption is relaxed and if possible and economical, one vehicle is allocated to more than one routes. With this assumption, the number of vehicles will be equal to the number of trips which naturally depend upon the capacity of the vehicles and the total delivery at all the delivery points.

In case the vehicle set comprises of vehicles of same type and capacity, then the problem is to determine optimal number of vehicles to be used for distribution. For obtaining the lower bound on the number of vehicles to be used, following formula can be used:

$$\text{Number of vehicles} = N = \sum_{j=2}^{n} D_j / C$$

where D_j → Load requirement at jth delivery point

C → Capacity of the vehicle

at $j = 1$, Demand is zero because it is supplying source

If N is not a whole number, it is rounded off to next higher whole number. The problem of designing the routes may be solved for several values of N and the best solution is chosen It has been experienced by solving some test problems that usually no more than three values of N need be considered (starting with lowest possible value of N and then increasing it in steps of one) for the final solution to be obtained.

Above formula holds good only in the case of same capacity vehicles. For different capacity vehicles, the cost of delivery, in addition to number of vehicles, depends upon the type of vehicles. Therefore, in such cases, the feasible

sets of vehicles for distribution are determined and problem may be solved for each of these sets to achieve the best solution. Vehicle-set-selection can be done by one of the following strategies:

(i) Vehicle-set-selection policy 1: Vehicles are picked from the available set of vehicles, for assignment, in descending order of their capacities.

(ii) Vehicle-set-selection policy 2: Vehicles are picked from the available set of vehicles, for assignment, in ascending order of their capacities.

(iii) Vehicle-set-solution policy 3: Vehicles are picked randomly from the available set of vehicles.

The minimum number of vehicles to be employed is r, such that

$$\sum_{i=1}^{r-1} C_i \leq \sum_{j=2}^{n} D_j$$

and

$$\sum_{i=1}^{r} C_i \geq \sum_{j=2}^{n} D_j .$$

where $D_j \rightarrow$ load requirement at jth customer

$C_i \rightarrow$ capacity of ith vehicle

Mathematically the various set selection policies can be represented as follows:

Policy I : $C_i > C_{i+1} > C_{i+2}$

Policy II : $C_i < C_{i+1} < C_{i+2}$

Policy III : C_i, selected randomly.

For each policy, problem is solved for r+s vehicles, where s = 0,1,2, ..., S; such that S+r = total number of available vehicles.

4.2.2 Designing the Routes

Having determined the sets of vehicles to be used, the problem of designing the routes is solved for each of these sets as follows:

(i) Construct the unconstrained initial routes, for minimizing the total distance travelled on the routes by vehicles.

(ii) Adjust the initial routes for satisfying constraints, increment in total distance, due to re-arrangement of customers on routes, is kept minimum.

(iii) Attempt to accomodate ommitted customers.

(iv) Improve the routes obtained by refining heuristics.

4.2.3 Construction of Initial Routes

Initial routes, one for each vehicle, are constructed to cover all the delivery points in the distribution network such that the sum of total distance of all the routes is minimum. While constructing initial routes, constraints are not imposed. Initial routes are developed by using Multiple travelling salesman algorithm. Multiple travelling salesman problem is the extension of single travelling salesman problem. In general, the objective of single travelling salesman proble is defined as to find a minimum cost (or distance) route for a salesman to visit each customer once and only once, starting from depot & coming back to depot. However, if the real

depot is eliminated and replaced by NV artificial depots (where NV is the number of vehicles used for distribution), all located in same position, then problem of constructing initial routes is similar to multiple travelling salesman problem and reads as 'Given NV vehicles and NC customers, find NV routes such that every customer is visited exactly once by exactly one vehicle, so that the total distance travelled by the vehicles is minimum.' An efficient algorithm for solving such problem is M-salesman travelling salesman algorithm(19). Initial routes have been obtained by this algorithm as follows.

As stated above, real depot is replaced by NV artific depots. Thus,

$$NN = NC + NV = (N-1) + (NV) = N + NV - 1$$

where N is total number of nodes in original network

NN is total number of nodes after replacement.

NV is number of vehicles.

The new distances $[C_{ij}]$ are obtained from the original distances $[d_{ij}]$ by augmenting the matrix $[d_{ij}]$ with (NV-1) new rows and columns, where each new row and column is a duplicate of the first row and column of the matrix $[d_{ij}]$ (because first row a column correspond to depot). All other new elements of the augmented matrix are set to infinity.

Using the notations used in Chapter 3, the problem for constructing initial routes can be structed similar to Miller-Tucker-Zemlin (27)'s formulation. Thus

$$\min Z = \sum_{i=1}^{NN} \sum_{j=1}^{NN} C_{ij}\, y_{ij} \qquad \ldots\ldots (1)$$

where Z is the total distance of all the routes.

Subject to

$$\sum_{i=1}^{NN} y_{ij} = 1, \qquad j = 1,2, \ldots, NN$$

$$\sum_{j=1}^{NN} y_{ij} = 1, \qquad i = 1,2, \ldots, NN \qquad \ldots\ldots (2)$$

$$y_{ij} = 0 \text{ or } 1 \text{ for all } i,j \qquad \ldots\ldots (3)$$

$$x_i - x_j + [\tfrac{N}{NV}]y_{ij} \leq [\tfrac{N}{NV}] + NV - 2 \qquad \ldots\ldots (4)$$

$$\text{for all } i \neq j \text{ and } i,j \notin I_o \text{ , } I_o = (1,2,\ldots,NV)$$

The variables x_i and x_j are arbitrary real numbers which satisfy the constraint (4). The braces [.] denote the largest integer not to exceed the value within the braces.

In above formulation, Equations (1), (2), (3) represent assignment problem formulation. The constraints (4) are the loop constraints which block the formation of infeasible subtours. Subtour is feasible only and only if it contains at least one artificial depot.

Stated in broad terms, the algorithm works as follows: An assignment problem with constraints (2) and (3) is solved. If the solution satisfies constraints (4), solution obtained, i.e., the initial routes constructed are optimal. If one or more of the constraints (4) is violated, a subset of the constraints (4) is implicitly introduced by setting certain elements of the distance matrix to infinity. The new problem is another assignment problem, which is solved and so on.

Hence, the algorithm, for generating initial routes, solves a sequence of assignment problems, which ends when the solution to one of the problems satisfies all the constraints (2), (3) and (4). The sequence of problems is defined in such a way as to assure that the terminal solution is optimal.

Briefly, the steps of the algorithm are as follows:

Step 1: Solve the associated assignment problem. If all subtours are feasible - terminate - the solution is optimal. If not, set Z_{LB} equal to the value of assignment problem. Construct a feasible solution by initial heuristics, set Z_{UB} (upper bound) equal to the value of the feasible solution and go to Step 2.

Step 2: Locate the shortest infeasible subtour in that assignment problem solution which defines Z_{LB}. Generate and solve k subproblems, one for each arc in the infeasible subtour. If the value of a subproblem $Z \geq Z_{UB}$, delete it from the set of subproblems, S, to be considered further. If $Z \leq Z_{UB}$ and the subproblem solution is feasible, redefine Z_{UB}, delte the subproblem, continue.

Step 3: If set S is empty, terminate, otherwise scan the set S for that problem such that $Z = \min_{i \in S} Z_i$ and set $Z_{LB} = Z$. If $Z_{LB} = Z_{UB}$ - terminate - otherwise go to Step 2.

4.2.4 Adjusting the initial routes for satisfying the constraints

In the previous section, unconstrained initial routes were obtained for the minimum total distance. These initial routes are now adjusted, to satisfy the constraints, by re-arranging the customers on the routes. Re-arrangement of the customers on the routes involves the increment in the total

distance of initial routes. Proposed algorithm, for adjustment of initial routes, attempts to re-arrange the customers on routes for adjustment, such that the increment in total distance of initial routes is minimized.

Algorithm starts with allocating the vehicles to the initial routes by following policies:

(a) Vehicle allocation policy 1: Allocate vehicles to routes in random order.

(b) Vehicle allocation policy 2: Allocate vehicles to routes in descending order of capacities, i.e., maximum capacity vehicle is allocated to first route and minimum capacity vehicle to last route.

(c) Vehicle allocation Policy 3: Allocate vehicles to routes in ascending order of capacities.

For each policy, allocation of vehicles to routes causes some routes to be overloaded* and some to be underloaded*. An attempt is made to accomodate customers from the overloaded routes on underloaded routes for minimal increment in distance due to reshuffling of customers. All the three ~~tree~~ policies listed above, are tested and the policy giving the best solution is chosen. For each of the policies, the routes are adjusted

---

*(1) Overloaded Route: If the total demand of customers on a route is more than capacity of vehicle allocated to it, is more, then route is called an overloaded route.

(2) Underloaded Route: If total demand of customers on a route is less than capacity of vehicle allocated, then route is called an underloaded route.

as follows:

Step 1: Firstly the proposed algorithm lists out all overloaded and/or overtimed routes* and adjusts one such route at a time. An immediate problem arises as to which overloaded route should be considered, because the order of presentation of overloaded routes might affect the solution. In order to observe the affect of order of presentation of overloaded routes for adjustment to algorithm, the following three route policies have been considered:

Route Policy 1: Overloaded route is selected randomly out of the overloaded routes.

Route Policy 2: Most overloaded route is selected first

Route Policy 3: Least overloaded route is selected first.

Overloaded route, obtained by a particular route policy, is adjusted as follows.

Step 2: Suppose the overloaded route to be considered for adjustment is OVR_i. Each customer on route OVR_i is considered to be discarded from route OVR_i and to be accomodated in some of other routes. Suppose customer A_1 is considered to be discarded from route OVR_i. This customer can be accomodated in any of other unadjusted routes and also to already adjusted routes if inclusion of it does not violate the capacity and time constraint of that adjusted route. Now a search is made to find the pair of linked nodes B_1 and C_1 on any of these permissible routes for accomodating A_1, such that new connections of A_1 with B_1 and C_1 involve the minimum distance increment MD_1. MD_1, B_1 and C_1 corresponding to discarded custome

* Overtimed route is the route for which route time is more than specified maximum route time.

A_1 are noted and customer A_1 is restored to route OVR_i. Then another customer A_2 is considered. Similarly for all the customers A_j on route OVR_i, search is made to find minimum distance incdrement MD_j and corresponding new connections for A_j, i.e., the pair of linked nodes B_j and C_j.

Step 3: In this step, a combination of customers on route OVR_i is sorted such that (i) removal of these customers adjusts the overloaded route under consideration, (ii) sum of minimum distance increments of discarded customers is minimized. This problem for sorting the best combination of customers, has been tackled by zero-one programming. Problem can be expressed in mathematical form as follows:

Let J-set of customers on overloaded route OVR_j

MD_j-minimum distance increment involved with removal of jth customer, where $j \varepsilon J$.

$OVC(i)$ - the overloaded quantity on route OVR_i.

$OVT(i)$ - the route time exceeded for route OVR_i

D_j - demand of jth customer, where $j \varepsilon J$

t_j - route time saving due to removal of jth customer from route OVR_i.

X_j - 0,1 variables.

X_j = 1 if jth customer is discarded
= 0 otherwise.

The combination of customers is to be chosen from the set J so as to,

$$\text{minimize} \sum_{j \varepsilon J} MD_j \, X_j \qquad \ldots\ldots (1)$$

$$\text{subject to} \sum_{j \varepsilon J.} t_j \, X_j \geq OVT(i)$$

$$\sum_{j \varepsilon J} D_j \, X_j \geq OVC(i) \qquad \ldots\ldots (2)$$

$$X_j = 0 \text{ or } 1 \qquad \ldots\ldots (3)$$

Zero-one programming solves the above problem and gives the optimal combination of customers to be discarded from route OVR_i in order to adjust it. The discarded customers are connected to their new connections B_j and C_j as found in Step 2. If any customer j on route OVR_i can't be included in any other route, MD_j is set to infinity so that this does not come into the solution giving the combination of custome: to be discarded. Such customers are kept aside and listed as ommitted customers.

Thus Step 3 adjusts the overloaded route OVR_i. This route is now called adjusted route for capacity and tim constraint and in further adjustments, customer can be accomodated to this route if and only if that customer does not violate the capacity and time constraints of this route.

Repeat the Step (1) through Step (3) till all the routes are adjusted for capacity and time constraints. Ommitted customers are also listed to be considered in next section.

4.2.5 Consideration of Ommitted Customers

Next, an attempt is made to accomodate the ommitted customers into some of the routes. Each ommitted customer is taken in turn and every route is inspected in order to determine whether by removing another customer, the ommitted one could be accepted; the customer removed is then fitted, if possible, into another route. If there is no customer left unassigned, we go ahead for improving the solution by refining heuristics. However, if some ommitted customers remain, we go back to Section 4.2.2 for considering another set of vehicles to be used for distribution.

4.2.6 Refining Heuristics

Routes have been improved further for reducing total delivery costs by the following refining heuristics.

(1) In first heuristic, each customer is considered in turn and the possibilities of moving him to all other positions in the same or another route are considered. He is moved to the position found for which the total distance is reduced and no constraints are violated.

(2) Second heuristic also attempts to reduce total distance. A customer from one route is placed in another route, if a customer from the second route could be inserted elsewhere in the system to reduce total distance. This procedure was developed because it was felt that the first heuristic might fail to place a customer in a route where this was desirable because of violation of capacity constraint. In such cases the room could have been made in the route by removing another customer from it by this heuristic.

(3) Lastly, a refining heuristic is applied for reducing the number of vehicles in use and hence reducing the total delivery cost. If some vehicles return to depot (after distribution) after only a few hours, it might be desirable to give a second trip to one or more vehicles if second trip can be completed within maximum allowed vehicle time. This procedure examines every pair of routes and searches for the pairs whose combined duration, after allowing for time to reload the vehicle, is below the maximum permitted time. This process does not reduce the number of distinct routes, therefore no savings in terms of total distance involved occur. But if two routes are combined and allocated to onevehicle, if possible, a vehicle is saved which in turn reduces the total delivery cost.

The computer programmes have been developed for all the algorithms discussed above. The computer package has been written in FORTRAN IV language and has been run on IBM 7044.

CHAPTER - V

COMPARISON OF PROPOSED MODEL WITH CLARKE & WRIGHT'S METHOD

In this chapter, the proposed model has been compared with the Clarke & Wright's (6) method for solving the vehicle dispatching problem. Nine test problems have been structured and solved by both the methods and the results obtained have been compared. Since Clarke & Wright s method does not take care of time constraint, the time constraint has not been considered in the test problems. The various costs of distribution are assumed to be the same for all the test problems. The various costs of distribution are tabulated in Table 5.1. The data on shortest distance between customers, customers' demand and the capacities of the vehicles for the various structured problems are given in tables 5.2 through 5.10. The shortest distance matrix has been generated by generating psuedo - random numbers between 0 and 1. The psuedo - random numbers have been generated by using the function RNDY5*. The shortest distance between the nodes i & j is determined using the relationship (d_{ij}) = (A * RNDY5) where A is some arbitrary chosen real number. For each problem a different value of A has been selected. Input data and results obtained from the Clarke & Wright's method and proposed model, for each test problem are given in Tables 5.2 through 5.10. Results indicate that in all the test problems proposed model has performed better in terms of total cost of delivery. A summary of the results obtained by the two methods is given in Table 5.11.

The results have been summarized in following paragraphs. The results indicate that

i) For each problem the total distance travelled by vehicles

* Note: RNDY5 is a built in function in IBM/7044.

to satisfy the demands of the set of customers considered in problem, is less in the case of the routes obtained by proposed method.

ii) The number of vehicles required for distribution by proposed method has been observed to be either equal to or less than found by the Clarke & Wright's method. In test problem numbers 2,3 and 9, the saving of a vehicle has been observed with proposed method.

iii) The cost of transportation is less with use of proposed method in all the test problems. Percentage saving in the cost of transportation with proposed method over Clarke & Wright's method varies from 6%, (for test problem 1) to 32% (for test problem 5).

iv) Total cost of delivery has been observed to be less in case of distribution by proposed method than that in case of the Clarke & Wright's method in all the test problems. Percentage saving in the total cost of delivery varies from 2% (for test problem 1) to 23% (for test problem 2).

v) Computational efficiency of the two methods has been compared in terms of execution time and computer memory needed. Table 5.11 shows the execution time for both the methods. Execution time for the proposed method has been observed to be higher than that for the Clarke & Wright's method for all the test problems. Computer memory requirement is also more for the proposed method than that required for Clarke & Wright's method.

To conclude the proposed method is not computationally as efficient as Clarke & Wright's method but this is more than made up by the fact that the proposed method results in considerable higher savings in cost to the tune of 10 to 20% and hence, is justifiably preferable.

Table 5.1

1) Operations Characteristics of vehicles

   i) Average speed = 20 Km./Hour

   ii) Consumption of fuel (petrol) 0.25 litre/km.

   iii) Fuel cost = Rs. 3.27/litre

2) Cost of transportation = Rs. 0.81/Km.

3) Maintenance cost per vehicle = Rs. 400 per month

4. Depreciation cost :

   Initial cost of vehicle = Rs. 80,000

   Depreciation rate (straight line method) = 10% per year.

   Depreciation cost per vehicle = Rs. 666.67 per month.

5) Labour cost :

   Salary to driver = Rs. 200/- per month

   Salary to helper = Rs. 175/- per month

   Salary to Route clerk = Rs. 250/- per month.

   Total Labour cost per vehicle = Rs. 800/- per month.

TABLE 5. 2

TEST PROBLEM 1

INPUT INFORMATIONS

NO. OF CUSTOMERS = 10
DISTANCE MATRIX GENERATED BY USING RNDY5
B(I,J) = IFIX(A.RNDY5(Y5))
A = 17.5
DEMANDS FOR CUSTOMERS
10 9 15 7 6 12 11 10 9 6
CAPACITY OF VEHICLES = 50

RESULTS

CLARKE AND WRIGHT METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	0	1 6 2 3 5 11 7 1
2	5	1 8 4 10 9 1

OPTIMAL NO. OF VEHICLES = 2
TOTAL DISTANCE= 67

COST OF TRANSPORTATION = 1641.50
MAINTENANCE,DEPRECIATION AND LABOUR COST= 3733.32
TOTAL COST= 5374.82

PROPOSED METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	4	1 3 9 7 10 6 1
2	1	1 4 2 5 11 8 1

BEST ROUTE POLICY IS 1

OPTIMAL NO. OF VEHICLES = 2
TOTAL DISTANCE= 63

COST OF TRANSPORTATION = 1543.50
MAINTENANCE,DEPRECIATION AND LABOUR COST= 3733.32
TOTAL COST= 5276.82

TABLE 5.3

TEST PROBLEM 2

INPUT INFORMATIONS

NO. OF CUSTOMERS = 13
DISTANCE MATRIX GENERATED BY USING RNDY5
B(I,J) = IFIX(A.RNDY5(Y5))
A = 10.0
DEMANDS FOR CUSTOMERS
23 25 24 18 25 20 24 22 18 19 20 17 26
CAPACITY OF VEHICLES = 100

RESULTS

CLARKE AND WRIGHT METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	76	1 4 1
2	15	1 6 9 5 7 1
3	13	1 8 11 14 10 1
4	15	1 12 2 13 3 1

OPTIMAL NO. OF VEHICLES = 4
TOTAL DISTANCE= 53

COST OF TRANSPORTATION = 1298.50
MAINTENANCE, DEPRECIATION AND LABOUR COST= 7466.64
TOTAL COST= 8765.14

PROPOSED METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	12	1 12 8 5 14 1
2	7	1 3 2 7 6 1
3	0	1 11 13 4 9 10 1

BEST ROUTE POLICY IS 1

OPTIMAL NO. OF VEHICLES = 3
TOTAL DISTANCE= 48

COST OF TRANSPORTATION = 1176.00
MAINTENANCE, DEPRECIATION AND LABOUR COST= 5599.98
TOTAL COST= 6775.98

T A B L E 5. 4

TEST PROBLEM 3

INPUT INFORMATIONS

NO. OF CUSTOMERS = 15
DISTANCE MATRIX GENERATED BY USING RNDY5
B(I,J) = IFIX(A.RNDY5(Y5))
A = 22.5
DEMANDS FOR CUSTOMERS
15 12 11 12 8 18 17 11 12 10 9 18 17 8 10
CAPACITY OF VEHICLES = 50

RESULTS

CLARKE AND WRIGHT METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	6	1 6 7 13 1
2	5	1 8 14 4 1
3	9	1 9 16 15 10 1
4	40	1 11 1
5	2	1 12 5 3 2 1

OPTIMAL NO. OF VEHICLES = 5
TOTAL DISTANCE= 130

COST OF TRANSPORTATION = 3185.00
MAINTENANCE, DEPRECIATION AND LABOUR COST= 9333.30
TOTAL COST= 12518.30

PROPOSED METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	7	1 15 14 13 1
2	0	1 10 3 7 6 1
3	3	1 8 4 11 12 1
4	2	1 2 16 5 9 1

BEST ROUTE POLICY IS 3

OPTIMAL NO. OF VEHICLES = 4
TOTAL DISTANCE= 108

COST OF TRANSPORTATION = 2646.00
MAINTENANCE, DEPRECIATION AND LABOUR COST= 7466.64
TOTAL COST= 10112.64

TABLE 5.5

TEST PROBLEM 4

```
INPUT INFORMATIONS
------------------

NO. OF CUSTOMERS =  18
DISTANCE MATRIX GENERATED BY USING RNDY5
          B(I,J) = IFIX(A.RNDY5(Y5))
A = 25.0
          DEMANDS FOR CUSTOMERS
30  60  70  40  35  32  48  23  32  41  35  58  35  31  37  42
44  35
CAPACITY OF VEHICLES =  175

          RESULTS
          -------

CLARKE AND WRIGHT METHOD
------------------------

ROUTE      UNUSED CAPACITY        ROUTE PATH
  1              2                 1 10  4  8  9  1
  2            134                 1 11  1
  3              6                 1 13  7 18  6  1
  4              2                 1 15 17  2 12 14  1
  5              3                 1 16  3  5 19  1

OPTIMAL NO. OF VEHICLES =  5
TOTAL DISTANCE= 170

COST OF TRANSPORTATION =  4165.00
MAINTENANCE,DEPRECIATION AND LABOUR COST=  9333.30
TOTAL COST= 13498.30

PROPOSED METHOD
---------------

ROUTE      UNUSED CAPACITY        ROUTE PATH
  1             12                 1 13 10 15 17  1
  2             22                 1  4  8 19  1
  3              2                 1  9  2 18  6 11  1
  4              8                 1 12  7  5  3  1
  5            103                 1 16 14  1

BEST ROUTE POLICY IS  2

OPTIMAL NO. OF VEHICLES =  5
TOTAL DISTANCE= 152

COST OF TRANSPORTATION =  3724.00
MAINTENANCE,DEPRECIATION AND LABOUR COST=  9333.30
TOTAL COST= 13057.30
```

TABLE 5. 6

TEST PROBLEM 5

INPUT INFORMATIONS

NO. OF CUSTOMERS = 20
DISTANCE MATRIX GENERATED BY USING RNDY5
B(I,J) = IFIX(A.RNDY5(Y5))
A = 27.5

DEMANDS FOR CUSTOMERS

25 18 16 23 30 12 20 18 20 15 20 15 17 23 21 20
20 15 16 16

CAPACITY OF VEHICLES = 100

RESULTS

CLARKE AND WRIGHT METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	2	1 4 16 20 11 9 7 1
2	2	1 12 21 2 17 14 1
3	12	1 13 8 5 6 1
4	4	1 18 15 3 10 19 1

OPTIMAL NO. OF VEHICLES = 4
TOTAL DISTANCE= 176

COST OF TRANSPORTATION = 4312.00
MAINTENANCE,DEPRECIATION AND LABOUR COST= 7466.64
TOTAL COST= 11778.64

PROPOSED METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	2	1 13 11 9 17 6 1
2	2	1 20 3 2 15 4 1
3	12	1 14 10 21 19 12 1
4	4	1 7 16 8 5 18 1

BEST ROUTE POLICY IS 2

OPTIMAL NO. OF VEHICLES = 4
TOTAL DISTANCE= 119

COST OF TRANSPORTATION = 2915.50
MAINTENANCE,DEPRECIATION AND LABOUR COST= 7466.64
TOTAL COST= 10382.14

T A B L E 5. 7

TEST PROBLEM 6

INPUT INFORMATIONS

NO. OF CUSTOMERS = 22
DISTANCE MATRIX GENERATED BY USING RNDY5
B(I,J) = IFIX(A.RNDY5(Y5))
A = 15.0

DEMANDS FOR CUSTOMERS

13 14 16 14 8 14 12 15 14 10 14 8 14 12 12 15
9 10 15 11 14 15

CAPACITY OF VEHICLES = 50

RESULTS

CLARKE AND WRIGHT METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	0	1 2 11 17 15 1
2	7	1 3 22 9 1
3	2	1 5 10 13 8 1
4	3	1 12 19 6 23 1
5	1	1 16 7 18 14 1
6	8	1 20 4 21 1

OPTIMAL NO. OF VEHICLES = 6
TOTAL DISTANCE= 124

COST OF TRANSPORTATION = 3038.00
MAINTENANCE,DEPRECIATION AND LABOUR COST= 11199.96
TOTAL COST= 14237.96

PROPOSED METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	6	1 7 10 4 1
2	2	1 13 8 14 5 1
3	0	1 21 20 19 12 1
4	3	1 3 17 11 6 1
5	2	1 16 18 15 9 1
6	8	1 2 23 22 1

BEST ROUTE POLICY IS 1

OPTIMAL NO. OF VEHICLES = 6
TOTAL DISTANCE= 110

COST OF TRANSPORTATION = 2695.00
MAINTENANCE,DEPRECIATION AND LABOUR COST= 11199.96
TOTAL COST= 13894.96

TABLE 5.8

TEST PROBLEM 7

INPUT INFORMATIONS

NO. OF CUSTOMERS = 25
DISTANCE MATRIX GENERATED BY USING RNDY5
B(I,J) = IFIX(A.RNDY5(Y5))
A = 20.0
DEMANDS FOR CUSTOMERS
30 60 70 40 35 32 48 23 32 41 35 58 35 31 37 42
44 35 25 28 28 32 28 31 32
CAPACITY OF VEHICLES = 200

RESULTS

CLARKE AND WRIGHT METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	19	1 6 24 7 21 19 9 1
2	13	1 8 16 10 4 1
3	18	1 12 3 20 23 2 1
4	17	1 14 17 15 25 18 1
5	1	1 26 11 5 13 22 1

OPTIMAL NO. OF VEHICLES = 5
TOTAL DISTANCE= 162

COST OF TRANSPORTATION = 3969.00
MAINTENANCE,DEPRECIATION AND LABOUR COST= 9333.30
TOTAL COST= 13302.30

PROPOSED METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	41	1 12 6 22 2 25 1
2	6	1 8 19 9 3 24 1
3	0	1 10 5 13 4 1
4	10	1 17 23 20 21 7 15 1
5	11	1 14 26 18 11 16 1

BEST ROUTE POLICY IS 2

OPTIMAL NO. OF VEHICLES = 5
TOTAL DISTANCE= 126

COST OF TRANSPORTATION = 3087.00
MAINTENANCE,DEPRECIATION AND LABOUR COST= 9333.30
TOTAL COST= 12420.30

TABLE 5.9

TEST PROBLEM 8

INPUT INFORMATIONS

NO. OF CUSTOMERS = 28
DISTANCE MATRIX GENERATED BY USING RNDY5
B(I,J) = IFIX(A.RNDY5(Y5))
A = 20.0
DEMANDS FOR CUSTOMERS
13 14 16 14 8 14 12 15 14 10 14 8 14 12 12 15
9 10 15 11 14 15 15 12 11 12 13 18
CAPACITY OF VEHICLES = 55

RESULTS

CLARKE AND WRIGHT METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	14	1 4 11 24 1
2	1	1 5 29 10 13 1
3	0	1 12 3 22 2 1
4	5	1 16 26 23 15 1
5	2	1 18 20 9 7 1
6	2	1 25 21 8 19 6 1
7	1	1 27 14 17 28 1

OPTIMAL NO. OF VEHICLES = 7
TOTAL DISTANCE= 205

COST OF TRANSPORTATION = 5022.50
MAINTENANCE,DEPRECIATION AND LABOUR COST= 13066.62
TOTAL COST= 18089.12

PROPOSED METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	2	1 18 14 13 11 8 1
2	2	1 21 24 17 25 1
3	3	1 7 9 20 6 1
4	3	1 4 16 26 2 1
5	2	1 22 23 15 27 1
6	9	1 10 29 12 1
7	4	1 5 19 3 28 1

BEST ROUTE POLICY IS 3

OPTIMAL NO. OF VEHICLES = 7
TOTAL DISTANCE= 142

COST OF TRANSPORTATION = 3479.00
MAINTENANCE,DEPRECIATION AND LABOUR COST= 13066.62
TOTAL COST= 16545.62

T A B L E 5.10

TEST PROBLEM 9

INPUT INFORMATIONS

NO. OF CUSTOMERS = 30
DISTANCE MATRIX GENERATED BY USING RNDY5
B(I,J) = IFIX(A.RNDY5(Y5))
A = 25.0

DEMANDS FOR CUSTOMERS

20 24 25 20 14 19 14 20 22 20 18 15 17 23 21 17
16 19 15 18 20 19 24 16 18 17 15 25 16 19

CAPACITY OF VEHICLES = 75

RESULTS

CLARKE AND WRIGHT METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	12	1 4 21 2 1
2	2	1 12 9 5 13 1
3	7	1 14 8 22 27 1
4	10	1 16 3 11 1
5	12	1 18 17 28 20 1
6	4	1 19 6 10 25 1
7	3	1 23 30 26 7 1
8	9	1 24 15 31 1
9	50	1 29 1

OPTIMAL NO. OF VEHICLES = 9
TOTAL DISTANCE= 229

COST OF TRANSPORTATION = 5610.50
MAINTENANCE,DEPRECIATION AND LABOUR COST= 16799.94
TOTAL COST= 22410.44

PROPOSED METHOD

ROUTE	UNUSED CAPACITY	ROUTE PATH
1	3	1 2 20 7 12 1
2	4	1 19 22 28 17 1
3	0	1 8 3 16 18 1
4	8	1 4 21 24 1
5	2	1 25 10 5 13 1
6	2	1 23 27 9 14 1
7	2	1 11 26 30 31 1
8	13	1 29 6 15 1

BEST ROUTE POLICY IS 2

OPTIMAL NO. OF VEHICLES = 8
TOTAL DISTANCE= 184

COST OF TRANSPORTATION = 4508.00
MAINTENANCE,DEPRECIATION AND LABOUR COST= 14933.28
TOTAL COST= 19441.28

TABLE 5.11

COMPARISON OF RESULTS - SUMMARY

Method I Clarke & Wright's Method

Method II Proposed method

Test Problem	No. of Customers	Total Distance		No. of vehicles required		Cost of transportation		Total cost of delivery		Computation time* (Secs.)	
		I	II	I	II	I	II	I	II	I	II
1	10	67	63	2	2	1641.50	1543.50	5374.82	5276.82	9	22
2	13	53	48	4	3	1298.50	1176.00	8765.14	6775.98	12	27
3	15	130	108	5	4	3185.00	2646.00	12518.30	10112.64	14	35
4	18	170	152	5	5	4165.00	3724.00	13498.30	13057.30	15	40
5	20	176	119	4	4	4312.00	2915.00	11778.64	10332.14	18	44
6	22	124	110	6	6	3038.00	2695.00	14237.96	13894.96	19	48
7	25	162	126	5	5	3969.00	3087.00	13302.30	12420.30	21	51
8	28	205	142	7	7	5022.50	3479.00	18089.12	16545.62	25	57
9	30	229	184	9	8	5610.50	4508.00	22410.44	19441.28	28	64

* Execution time only.

CHAPTER VI

VEHICLE DISPATCHING FOR KANPUR SAHAKARI MILK BOARD - A CASE STUDY

6.1 The proposed model has been used for the development of distribution routes and vehicle dispatching strategies for Kanpur Sahakari Milk Board which is engaged in distribution of milk products in Kanpur City. The company has a wide distribution network and a fleet of delivery trucks. A brief description of the operations of Kanpur Sahakari Milk Board (KSMB). is given in the following paragraphs:

6.1.1 Procurement Raw milk is collected at various collection centres having milk potentiality. At some of these collection centres where the milk potentiality is high, company has got its chilling centres for preserving the collected milk. Presently three company owned tankers and six hired private vehicles transport this collected milk to the Dairy for processing.

6.1.2 Processing: Procured raw milk and imported milk powder is processed to yield pasteurized milk products namely Milk, butter, ghee and cream etc. Processing is done for preparing tonned, double tonned and skimmed milk etc. having different fat percentages. Processed milk is bottled in half-litre capacity bottles and kept in cold storage, ready for distribution.

6.1.3 Distribution: The Company has a wide distribution network and a fleet of delivery trucks. The milk products are distributed to the general public through the agents at various delivery centres. In other words, Distribution Channel followed is Producer to retailer to consumer. Company is distributing the products twice a day (morning & evening) to about 250 agents in Kanpur city who sell them to public. Therefore, so far the company's distribution

system is concerned, agents are the customers. Main products to be delivered are milk and butter. For Ghee, there are only two delivery centres, one at the dairy itself and other at Super Market, Kanpur. There are just two or three deliveries per month to Super Market, therefore the distribution of Ghee has not been scheduled in the regular distribution system. Presently, the company is engaging Nine delivery trucks for distribution, seven out of them are bigger (Capacity of 4000 bottles) and two smaller (capacity of 1600 bottles). All the nine trucks are allocated to 9 routes in the morning shift and six trucks (5 bigger and 1 smaller) trucks for 6 routes in evening shift, demand in evening being lesser.

Delivery trucks are loaded with crates (each containing 20 bottles) at the loading centre. The loaded truck is accompanied by two helpers, one route clerk and obviously one driver for the distribution of products. Helpers load and unload the crates at the delivery points and depot. Route clerk is responsible for the proper distribution and money transaction etc. Full delivery to each customer is being made simultaneously. At each customer point, the crates of returned empty bottles are loaded in truck. After delivering to every customer on the routes, trucks return to the dairy (depot) and all the crates of empty bottles in truck are unloaded. This completes the distribution cycle.

For conducting the case study, various data need to be collected as an input information to the model. Following section discusses about the data collection for the use of present study.

6.2 <u>Data Collection</u>: The strategy used for the collection of data has been discussed in this section. Mainly five set of data are required viz.:

a) Distances between the delivery points,

b) Demands at various delivery points,

c) Operating characteristics of available delivery vehicles,

d) Costs involved in distribution.

Before going ahead, it is opportune to discuss some procedural simplifications made, for making the study convenient, as follows:

6.2.1 Aggregation of delivery points

The number of delivery points at which milk products are distributed in Kanpur City is around two hundred and fifty. Because of computational difficulties and limited computer memory, it is not possible to consider them individually. Therefore, delivery points have been grouped. A group of customers are supposed to be supplied from a single delivery point. Firstly the delivery points in a locality have been grouped and each group has been represented by a delivery point in that locality. These delivery points have been plotted on the map of Kanpur City. These delivery points turned out to be 88 in number. Since the proposed model can handle about 30 customers because of computer storage limitations on IBM/7044, further grouping of these delivery points was needed. The plotted delivery points forming the clusters have been grouped first. The delivery points which do not form the clusters have been grouped, based on proximity criterion. The delivery points within a diameter of 1.5 km have been considered to be at a single delivery point. By this proximity criterion twenty three groups have been formed. But it was observed that the number of individual delivery points in some groups, which were in densly populated area, was quite high as compared to that in thinly populated area. Therefore, second criterion was used for grouping. This criterion groups the indiviaual points in densly populated area within diameter of 1 km which yields the number of delivery points in each group more or less same i.e., between 8 to 10 delivery points, in each group. Thus, in nutshell, the criteria used for aggregation of delivery points can be spelled out as 'the delivery points within a diameter of 1.5 km in the densely populated area and delivery points within 1.0 km in a thinly populated area are considered to be at a single delivery point which has been named as zonal delivery point.' Above procedure of aggregation of delivery points yields Twenty seven zonal delivery points which have been considered as the nodes in the distribution network

for the present case study. These nodes have been listed in table 6.1. The total requirement of individual delivery points group, named as zone is represented as the load at zonal delivery point of that zone. The total delivery to the delivery in any zone is supposed to be delivered at the zonal delivery point.

6.2.2 Milk as the only product to be delivered: In the present distribution system, among the milk products which are distributed, milk has been considered as the main product for distribution. The ghee and butter are needed in smaller quantities and are being distributed at fewer delivery centres. Moreover, the capacity contribution of these products to the vehicles for distribution being very small, these products have not been considered for distribution. This assumption leave milk as the only product for distribution in our study.

6.3 The strategies used for the collection of data are outlined in the following paragraphs:

6.3.1 Distances between the nodes in distribution network: As discussed in Chapter 3, the shortest distances between the nodes are required as an input data to the model. Shortest distances are obtained from direct linked distances as discussed in Chapter 4, by shortest path algorithm. Therefore, data have been collected to prepare direct linked distance matrix showing the distances of direct path between every pair of nodes. For collecting these data, the zonal delivery points and milk depot are plotted on the road map of the Kanpur city. The road distances between each pair of zonal delivery points, which are connected by road are determined. If there are many possible direct links between two zonal delivery points, the shortest direct link is considered. The unit of distance matrix is one forth of a kilometer. Wherever there is no direct link between two delivery points distance between them has been assumed as infinity. Distance matrix has been shown in Table 6.2(a). Shortest distance matrix, as computed by shortest path algorithm, has been shown in Table 6.2(b).

TABLE 6.1

NODE	ZONE-NAME
1	Milk Board
2	Khapra Mohal
3	Generalganj
4	Ram Narain Bazar
5	Chowk
6	Birhana Road
7	Cantonment
8	Latouch Road
9	Chamanganj
10	Anwarganj Station
11	Faithfulganj
12	Govindnagar (East)
13	Govindnagar (West)
14	Ordinance factory
15	Shastrinagar
16	Lajpat Nagar
17	L.L.R. Hospital
18	IIT
19	Aryanagar
20	Nawabganj
21	Azadnagar
22	Gwaltoli (South)
23	Gwaltoli (North)

NODE	ZONE NAME
24	Fazalganj
25	Jawaharnagar
26	Kidwainagar
27	Civil Lines
28	Parade

TABLE 6.2 (b)

SHORTEST DISTANCE MATRIX

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	0	37	32	33	31	36	36	27	22	18	32	6	6	24	22	17	25	65	26	37	44	30	35	10	18	14	30	27
2	37	0	8	7	10	6	8	10	20	19	8	37	37	47	41	34	33	73	25	36	43	20	21	30	22	24	16	13
3	32	8	0	4	4	5	16	5	12	14	12	32	32	42	35	28	25	65	17	28	35	12	13	22	14	25	12	5
4	33	7	4	0	3	4	15	8	13	17	13	35	35	45	36	29	26	66	18	29	36	13	14	23	15	28	11	6
5	31	10	4	3	0	7	18	7	11	16	10	33	34	44	34	27	24	64	16	27	34	11	12	21	13	27	8	4
6	36	6	5	4	7	0	14	9	16	18	9	36	36	46	39	32	29	69	21	32	39	16	17	26	18	27	12	9
7	36	8	16	15	18	14	0	18	28	26	12	37	42	54	48	41	40	80	33	44	51	28	29	38	30	22	24	21
8	27	10	5	8	7	9	18	0	10	9	17	27	27	37	31	24	23	63	22	33	40	17	18	25	19	20	15	10
9	22	20	12	13	11	16	28	10	0	10	21	22	24	38	32	25	22	62	14	25	32	14	15	18	12	25	14	7
10	18	19	14	17	16	18	26	9	10	0	14	18	18	28	22	15	14	54	18	26	33	24	25	16	12	20	24	17
11	32	8	12	13	10	9	12	17	21	14	0	32	32	42	36	29	28	68	26	37	44	21	22	30	23	18	18	14
12	6	37	32	35	33	36	37	27	22	18	32	0	10	28	18	18	26	66	32	38	45	36	37	16	24	15	36	29
13	6	37	32	35	34	36	42	27	24	18	32	10	0	18	16	20	28	68	32	40	47	36	39	16	24	20	36	31
14	24	47	42	45	44	46	54	37	38	28	42	28	18	0	19	26	32	72	40	44	51	48	53	28	36	38	48	45
15	22	41	35	36	34	39	48	31	32	22	36	18	16	19	0	7	13	53	21	25	32	31	38	14	21	33	33	30
16	17	34	28	29	27	32	41	24	25	15	29	18	20	26	7	0	8	48	16	20	27	26	31	7	14	31	26	23
17	25	33	25	26	24	29	40	23	22	14	28	26	28	32	13	8	0	40	8	12	19	18	25	15	11	34	23	20
18	65	73	65	66	64	69	80	63	62	54	68	66	68	72	53	48	40	0	48	52	59	58	65	55	51	74	63	60
19	26	25	17	18	16	21	33	22	14	18	26	32	32	40	21	16	8	48	0	11	18	10	17	16	8	38	18	12
20	37	35	28	29	27	32	44	33	25	26	37	38	40	44	25	20	12	52	11	0	7	21	28	27	19	46	29	23
21	44	43	35	36	34	39	51	40	32	33	44	45	47	51	32	27	19	59	18	7	0	28	35	34	26	53	36	30
22	30	20	12	13	11	16	28	17	14	24	21	36	36	48	31	26	18	58	10	21	28	0	7	20	12	37	10	7
23	35	21	13	14	12	17	29	18	15	25	22	37	39	53	38	31	25	65	17	28	35	7	0	25	17	38	8	8
24	10	30	22	23	21	26	38	25	18	16	30	16	16	28	14	7	15	55	16	27	34	20	25	0	8	24	20	17
25	18	22	14	15	13	18	30	19	12	12	23	24	24	36	21	14	11	51	8	19	26	12	17	8	0	32	12	9
26	14	24	25	28	27	27	22	20	25	20	18	15	20	38	33	31	34	74	38	46	53	37	38	24	32	0	35	30
27	30	16	12	11	8	12	24	15	14	24	18	36	36	48	33	26	23	63	18	29	36	10	8	20	12	35	0	7
28	27	13	5	6	4	9	21	10	7	17	14	29	31	45	30	23	20	60	12	23	30	7	8	17	9	30	7	0

TABLE 6.2 (a)

DISTANCE MATRIX

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	0	999	999	999	999	999	999	999	22	18	999	6	6	999	999	999	999	999	999	999	999	999	999	10	999	14	999	999
2	999	0	8	7	10	6	8	10	999	999	8	999	999	999	999	999	999	999	999	999	999	999	999	999	999	24	16	13
3	999	8	0	4	4	5	999	5	999	999	12	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	5
4	999	7	4	0	3	4	999	8	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	11	6
5	999	10	4	3	0	9	999	7	999	999	10	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	8	4
6	999	6	5	4	9	0	999	9	999	999	9	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	12	9
7	999	8	999	999	999	999	0	999	999	999	12	999	999	999	999	999	999	999	999	999	999	999	999	999	999	22	32	999
8	999	10	5	8	7	9	999	0	10	9	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	20	999	10
9	22	999	999	999	999	999	999	10	0	10	999	22	24	999	999	999	999	999	14	999	999	16	999	18	12	25	999	7
10	18	999	999	999	999	999	999	9	10	0	14	18	18	28	26	15	14	999	18	999	999	999	999	16	12	20	999	999
11	999	8	12	999	10	9	12	999	999	14	0	999	999	999	999	999	999	999	999	999	999	999	999	999	999	18	20	16
12	6	999	999	999	999	999	999	999	22	18	999	0	10	999	18	18	999	999	999	999	999	999	999	18	999	15	999	999
13	6	999	999	999	999	999	999	999	24	18	999	10	0	18	16	20	999	999	999	999	999	999	999	20	999	999	999	999
14	999	999	999	999	999	999	999	999	999	28	999	999	18	0	19	28	999	999	999	999	999	999	999	28	999	999	999	999
15	999	999	999	999	999	999	999	999	999	26	999	18	16	19	0	7	13	999	999	999	999	999	999	14	999	999	999	999
16	999	999	999	999	999	999	999	999	999	15	999	18	20	28	7	0	8	999	999	999	999	999	999	7	14	999	999	999
17	999	999	999	999	999	999	999	999	999	14	999	999	999	999	13	8	0	40	8	12	999	999	999	999	11	999	999	999
18	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	40	0	999	999	999	999	999	999	999	999	999	999
19	999	999	999	999	999	999	999	999	14	18	999	999	999	999	999	999	8	999	0	11	999	10	999	999	8	999	18	12
20	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	12	999	11	0	7	999	999	999	999	999	999	999
21	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	7	0	999	999	999	999	999	999	999
22	999	999	999	999	999	999	999	999	16	999	999	999	999	999	999	999	999	999	10	999	999	0	7	999	12	999	10	7
23	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	999	7	0	999	999	999	8	8
24	10	999	999	999	999	999	999	999	18	16	999	18	20	28	14	7	999	999	999	999	999	999	999	0	8	999	999	999
25	999	999	999	999	999	999	999	999	12	12	999	999	999	999	999	14	11	999	8	999	999	12	999	8	0	999	12	9
26	14	24	999	999	999	999	22	20	25	20	18	15	999	999	999	999	999	999	999	999	999	999	999	999	999	0	999	999
27	999	16	999	11	8	12	32	999	999	999	20	999	999	999	999	999	999	999	18	999	999	10	8	999	12	999	0	7
28	999	13	5	6	4	9	999	10	7	999	16	999	999	999	999	999	999	999	12	999	999	7	8	999	9	999	7	0

NOTE- D(I,J) MEANS SHORTEST DIRECT LINK BETWEEN I AND J NODE
D(I,J)=999 MEANS NO DIRECT LINK BETWEEN I AND J NODE

6.3.2 Demand at each delivery point: Demands at zonal delivery points are the input data to the model. Demand at any zonal delivery point is the summation of the demands of all delivery points grouped in that zone. Therefore, the demand at each delivery point is found first. Demand at any delivery point is the quantity indented by the agent at that delivery point. This quantity indented by the agent can be obtained from the indent book available with Sales Office of the Milk Board. Company is distributing milk twice a day (both in the morning and in the evening). Therefore, the morning and evening demand data at each delivery point have been collected first. With these data, the average morning and evening demand at each delivery point is calculated for each month. Indent book was available from January, 1974 only. So, because of non-availability of enough demand data, no forecasting model could be tried for estimation of demands. However it was observed that morning and evening demand at each delivery point were not significantly varying for January, February and March months. Therefore, the average morning and evening demand for these three months were found. These average demands at delivery points have been called as Winter morning and evening demands. Similarly, the average morning and evening demands in months April, May and June haave been found and have been called Summer morning and evening demands at delivery points. These average demands of all the delivery points in each zone are summed up to give average morning and evening demand (for summer and winter separately) at that zonal delivery point. Table 6.3 indicates the average demands at zonal delivery points.

6.3.3 Operating Characteristics of Vehicles

As was mentioned in Chapter 3, all vehicles are in same condition. So operating characteristics depend upon the size of vehicle and not on the condition of the vehicle. For each vehicle, the average travelling speed and maximum allowable load i.e. capacity is noted. These data have been tabulated in Table 6.5.

TABLE 6.3

DEMAND DATA (KANPUR SAHAKARI MILK BOARD)

NODE	SUMMER		WINTER	
	MORNING	EVENING	MORNING	EVENING
1	0	0	0	0
2	720	475	610	360
3	650	435	550	320
4	750	500	620	390
5	770	500	650	400
6	930	660	680	510
7	1100	640	930	640
8	1040	620	920	620
9	1150	710	930	600
10	500	335	340	160
11	740	580	620	380
12	920	620	810	510
13	670	425	550	310
14	820	580	730	480
15	600	435	500	300
16	760	510	650	380
17	1400	1165	1300	1000
18	800	810	500	280
19	1500	1070	1250	800
20	870	560	710	480
21	670	460	650	400
22	960	620	810	530
23	1040	680	870	600
24	1030	700	890	550
25	1140	760	1020	670
26	870	550	750	450
27	1200	760	1040	700
28	700	650	650	400

TABLE 6.4

VEHICLE CAPACITIES

Number of available vehicles = 9

Vehicle number	Capacity (bottles) *
1	4000
2	4000
3	1600
4	4000
5	1600
6	4000
7	4000
8	4000
9	4000

*Each bottle of half litre capacity.

TABLE 6.5

OPERATING CHARACTERISTICS OF VEHICLES AND DISTRIBUTION COSTS INVOLVED

i) Average stop-off-time at each individual agent = 3.5 minutes

ii) Average stop-off-time at depot = 120 minutes

iii) Maximum Permissible route time = 300 minutes

iv) Maximum Permissible Vehicle time = 420 minutes

v) Average fuel consumption of diesel for vehicles with capacity of 4000 bottles = 4 km. per litre

vi) Average fuel consumption of petrol for for small-vehicles = 5 km/litre.

vii) Average speed of bigger vehicles (capacity 4000 bottles) in morning = 20 k.m. per hour

viii) Average speed of bigger vehicles in evening = 15 km. per hour

ix) Average speed of smaller vehicles (capacity 1600 bottles) in morning = 25 km. per hour

x) Average speed of smaller vehicles in evening = 20 km. per hour

xi) Maintenance cost per vehicle per shift = Rs. 167 per month

xii) Depreciation cost* for bigger vehicle = Rs. 277 per month

xiii) Depreciation cost* for smaller vehicle = Rs. 84 per month

xiv) Salaries

Drivers salary = Rs. 250/- p.m.

Route clerk's salary = Rs. 275/- p.m.

Helper's Salary = Rs. 210/- p.m.

Total Labour cost = Rs. 945/- p.m.

xv) Cost of petrol = Rs. 3.27 per litre

Cost of diesel = Rs. 1.08 per litre

* Annual depreciation cost is 10% of initial purchase cost.

6.3.4 Distribution costs involved: Average consumption of fuel for each vehicle has been found. Knowing it, cost of fuel per kilometer for each vehicle has been found. Labour Cost includes Salary of driver, helpers and route clerk. These data have been taken from the accounts office of the milk board. Maintenance and depreciation costs for each vehicle on monthly average basis have been taken for each vehicle from the transport section of the milk board. Table 6.5 shows the different costs involved in distribution.

6.3.5 Stop-off-time at milk depot and delivery points: Estimates of stop-off-time* at the depot and the delivery points have been obtained from the dispatching section, Kanpur Sahakari Milk Board. Stop-off-time at a zonal delivery point, which is required as input data to Model, has been taken as summation of average stop-off-time at all the delivery points in that zone. Table 6.6 shows the stop-off-times at the zonal delivery points.

* average stop-off-time

TABLE 6.6

NUMBER OF AGENTS AND STOP-OFF-TIME AT THE NODES

NODE	NO. OF AGENTS	NO. OF AGENTS (ACTUALLY SERVED)		STOP OFF TIME (MINUTES)	
		MORNING	EVENING	MORNING	EVENING
1	0	0	0	120	120
2	9	9	6	31	21
3	10	10	6	35	21
4	9	9	5	31	17
5	8	7	4	25	14
6	10	9	6	31	21
7	10	10	6	28	21
8	9	9	5	31	17
9	9	9	5	31	17
10	10	10	7	35	25
11	9	9	6	31	21
12	8	8	6	28	21
13	10	10	7	35	25
14	9	9	4	31	14
15	10	10	8	35	28
16	8	8	7	28	25
17	9	9	9	31	31
18	10	10	4	35	14
19	10	9	7	31	25
20	10	10	7	35	25
21	8	8	5	28	17
22	10	10	6	35	21
23	10	10	7	35	25
24	10	10	7	35	25
25	9	9	5	31	17
26	11	11	7	38	25
27	7	7	6	25	21
28	9	9	8	31	28

CHAPTER-VII

RESULTS & DISCUSSION

In this chapter, the results obtained for the distribution system of the Kanpur Sahakari Milk Board have been presented and discussed. The distribution policies obtained using the model have been compared with the distribution policies presently adopted by the management of the Kanpur Sahakari Milk Board. A sensitivity analysis has been conducted for observing the sensitivity of results to the various parameters. Recommendations for the further work are given at the end of this Chapter.

7.1 Results for the distribution system of the Kanpur Sahakari Milk Board. The proposed distribution routes and vehicle dispatching strategies for the Kanpur Sahakari Milk Board presented in this section. The best possible results obtained by the proposed model, for the morning and evening shifts of the summer and winter months, are represented in tables 7.1 and 7.2, respectively.Table 7.1 indicates that the proposed vehicles, for the distribution in morning shift of summer months, are the vehicle numbers 1,2,3,4, 6,7,8*. This table also indicates the route path for each route and the vehicle allocation to the routes. Untilized capacities of vehicles (UVC) are given in column 4. The time (RT) for which vehicles are on the routes and the time for which each vehicle is engaged for distribution is indicated in column 5. The cost of transportation and the total cost, corresponding to the proposed routes and vehicle dispatching strategies for morning shift of summer months, are Rs. 1,239.77. per month and Rs. 10,768.77 per month respectively, as indicated by the table 7.1.(a)

The results for the morning and evening shifts of winter months are tabulated in table 7.2 which indicates the route path for the routes obtained and allocation of the selected vehicles to the routes. Costs of transportation for morning and evening shift are Rs. 1,407.10 per month and Rs. 1,168/- per month respectively. Total distribution costs are Rs. 9 547.10 per month and Rs. 7,725/- per month respectively, as indicated by table 7.2(a) and 7.2 (b).

*The available vehicles have been numbered as indicated in table 6.4

TABLE 7.1(a)

FOR SUMMER-MORNING DEMANDS

Route	Vehicle Allocation		Unused capacity (UVC)	Route time (RT) Mints.	Vehicle Time (VT) Mints.	Route Path
	Number	Capacity (bottles)				
1	3	1600	10	76	196	1-13-12-1
2	1	4000	270	188	308	1-8-2-7-26-1
3	2	4000	100	185	305	1-27-23-22-28-1
4	4	4000	160	255	375	1-18-21-20-19-1
5	6	4000	180	181	301	1-25-9-10-24-1
6	7	4000	160	216	336	1-11-6-4-5-3-1
7	8	4000	420	185	305	1-14-15-17-16-1

Cost of transportation = Rs. 1,239.77 per month

Total cost of distribution = Rs. 10,768.77 per month

TABLE 7.1(b)

FOR SUMMER-EVENING DEMANDS

Route	Vehicle-allocation Number	Vehicle-allocation Capacity (bottles)	Unused capacity (UVC)	Route time (RT) Mints.	Vehicle time (RT) Mints.	Route path
1	3	1600	140	69	189	1-25-24-1
2	1	4000	120	259	379	1-19-20-21-22-11-26-1
3	2	4000	75	293	413	1-16-17-18-15-14-13-1
4	4	4000	335	233	353	1-7-2-6-4-3-8-10-1
5	6	4000	80	215	335	1-27-23-5-28-9-12-1

Cost of transportation = Rs. 1,107.00 per month.

Total cost of distribution = Rs. 7,858.00 per month.

TABLE 7.2(a)

FOR WINTER-MORNING DEMANDS

Route	Vehicle-allocation Number	Capacity (bottles)	Unused Capacity (UVC)	Route time (RT) Mints.	Vehicle Time (VT) Mints.	Route Path
1	1	4000	200	258	378	1-4-6-2-7-11-10-1
2	2	4000	140	228	348	1-26-3-5-23-27-
3	4	4000	250	180	300	1-13-9-19-25-1
4	6	4000	190	283	403	1-21-20-18-17-16-1
5	3	1600	140	105	225	1-28-22-1
6	7	4000	150	245	365	1-24-15-14-8-12-1

Cost of transportation = Rs. 1,407.10 per month

Total cost of distribution= Rs. 9,547.10 per month

TABLE 7.2(b)

FOR WINTER-EVENING DEMANDS

Route	Vehicle-allocation Number	Capacity (bottles)	Unused capacity (UVC)	Route time (RT) Mints.	Vehicle time (VT) Mints.	Route Path
1	3	1600	60	207	327	1-18-20-21-16-1
2	5	1600	50	94	214	1-17-24-1
3	1	4000	330	263	383	1-9-25-19-15-14-13-12-1
4	2	4000	780	251	371	1-28-5-2-6-4-3-8-10-1
5	4	4000	700	245	365	1-26-7-11-27-23-22-1

Cost of transportation = Rs. 1,168.00 per month

Total cost of distribution = Rs. 7,725.00 per month

7.2 Comparison of the total distribution cost by proposed dispatching policies with that of the currently used policies of Kanpur Sahakari Milk Board: The proposed distribution routes and the vehicle dispatching strategies for the Kanpur Sahakari Milk Board have been discussed in Section 7.1 and are tabulated in Tables 7.1 and 7.2. The current distribution costs for the existing distribution system of Kanpur Sahakari Milk Board, are given in table 7.3. Table 7.4 shows the comparison of the distribution cost by the proposed vehicle policies with that of the currently used policies of Kanpur Sahakari Milk Board. The savings in total distribution cost obtained by the use of proposed model, are indicated in last column of Table 7.4. For the monring shift of summer months, the saving of 24% has been observed. For the evening shift of summer months, the saving has been observed to the tune of 21% .

For the morning and evening shifts of winter months, the savings in total distribution costs have been observed more than that for summer months. The reason for the higher savings in winter months can be explained as follows: The number of vehicles (obtained by proposed model) employed for satisfying customers' demands in winter months, are less than that employed in summer months, because of less demand in winter months. On the other hand, the number of vehicles employed for the distribution, by the management, Kanpur Sahakari Milk Board, are same for both winter and summer months. For the morning shift of winter months, the saving obtained by proposed solution is 32.6%. For the evening shift of winter months, the saving observed is 22.8%. In general, the savings obtained by the proposed distribution policies over the currently used policies by Kanpur Sahakari Milk Board are to the tune of 20 to 32%.

7.3 Effect of various vehicle-set-selection policies and vehicle-allocation policies : In this section, the effect of various vehicle-set selection policies and vehicle allocation policies on the objective function has been discussed. To study the effect of various policies on the objective function, the problem has been solved with various policies

TABLE 7.3

DISTRIBUTION COSTS FOR EXISTING DISTRIBUTION SYSTEM OF KANPUR SAHAKARI MILK BOARD

(Estimates given by Management, Kanpur Sahakari Milk Board)

i) For morning shift

Cost of transportation = Rs. 2,100.00 per month

Maintenance, labour and depreciation cost = Rs. 12,113.00 per month

Total cost = Rs. 14,213.00 per month.

ii) For evening shift

Cost of transportation = Rs. 1,797.00 per month.

Maintenance, labour and depreciation cost = Rs. 8,140.00 per month

Total cost of distribution = Rs. 9,937.00 per month

TABLE 7.4

COMPARISON OF THE TOTAL DISTRIBUTION COST BY PROPOSED DISPATCHING POLICIES WITH THAT OF THE CURRENTLY USED POLICIES OF KANPUR SAHAKARI MILK BOARD

	Total cost of distribution (Rs/month)		Savings %
	Existing System	Proposed System	
Summer-morning	14213.00	10768.77	24.0%
Summer evening	9937.70	7858.00	21.2%
Winter-morning	14213.00	9547.10	32.6%
Winter-evening	9937.70	7725.00	22.8%

for the different sets of demands. Different sets of demands have been obtained by variating the original demands at each customer from -20% to +20% in step of 5%. For each demand set, the total costs of distribution for different vehicle set-selection-policies and vehicle-allocation policies ahve been indicated in Table 7.5. These results indicate that different combinations of vehicle-set-selection policies and vehicle-allocation policies yield different costs of distribution.

7.3.1 Effect of Vehicle-set-selection policies

Vehicle-set-selection policy selects the set of vehicles for distribution to satisfy the customers' demands. In this section, the effect of different vehicle-set-selection policies on the cost of distribution has been discussed. Fig. 1 shows the maintenance, depreciation and labour costs for different vehicle-set-selection policies with variation in summer-morning demands of each customer from -20% to +20% in step of 5%. Fig. 1 shows that vehicle-set-selection policy 2, in general, gives higher cost, exceptional cases being only when demand at each node increases by 5% or reduces by 10%. Use of vehicle-set-selection policy 2 results in higher cost because in this policy, the available vehicles are arranged in ascending order of their capacities and smaller capacity vehicles are selected first. Therefore for satisfying the customer demands, vehicle-set-selection plicy2 requires more number of vehicles than required by vehicle-set-selection policy 1 in which vehicles are arranged in decending order of the capacities. When demand increases by 5%, the reason for lower cost by using policy 2 than that by using policy 3, is that policy 2 selects 2 small and 6 big capacity vehicles whereas the policy 3 selects 1 small and 7 big capacity vehicles. The number of vehicles in both cases are same but higher depreciation cost, in case of big capacity vehicles, causes policy 3 to yield higher cost.

In Fig. 2 a graphical relationship between the total cost of distribution and the various demand sets considering different vehicle-set-selection policies is presented. Fig. 2 shows that in all the cases, vehicle-set selection policy 2 gives higher cost. When demand is varied by

TABLE 7.5

TOTAL COST OF DISTRIBUTION (Rs./MONTH) FOR DIFFERENT SETS OF DEMANDS, USING VARIOUS SET SELECTION POLICIES AND VEHICLE ALLOCATION POLICIES

(Different sets of demands have been obtained by variating the summer-morningdemands in step of $\pm 5\%$)

Per-centage Varia-tion	Total demands (bottles/day)	TOTAL COST OF DISTRIBUTION (Rs./MONTH)								
		Vehicle-set-selection Policy 1			Vehicle-set-Selection Policy 2			Vehicle-set-Selection Policy 3		
		Vehicle-allocation policies			Vehicle-allocation policies			Vehicle-allocation Policies		
		I	II	III	I	II	III	I	II	III
-20	19440	9815	9815	9815	10904	10894	10920	9540	9592	9732
-15	20655	9755	9755	9755	10998	11018	11251	9550	9321	9435
-10	21870	9509	9509	9509	11031	10901	10991	10712	10709	10796
-5	23085	9524	9524	9524	12209	12100	12290	10865	10752	10710
0	24300	10990	10990	10990	12117	12154	12307	10914	10873	10768
5	25515	11183	11183	11183	12283	12318	12398	12181	12095	12219
10	26730	11098	11098	11098	13480	13310	13430	12410	12501	12198
15	27945	12398	12410	12480	13570	13400	13510	12218	12248	12338
20	29000	12518	12472	12540	13498	13512	13408	12518	12472	12540

TABLE 7.6

Cost of maintenance, depreciation and labour cost (Rs./Month), for different sets of demands, using various set policies. (Different sets of demands have beeb obtained by variating the summer-morning-demands in Step of ±5%)

Percentage Variation	Total demands (bottles/ day)	Cost of maintenance, depreciation & labour cost (Rs./month)		
		Vehicle-set-selection Policty 1	Vehicle-set-selection Policy 2	Vehicle-set-selection Policy 3
-20	19440	8334	9335	8140
-15	20655	8334	9335	8140
-10	21870	8334	9335	9529
-5	23085	8334	10724	9529
0	24300	9723	10724	9529
5	25515	9723	10724	10918
10	26730	9723	12113	10918
15	27945	10918	12113	10918
20	29100	10918	12113	10918

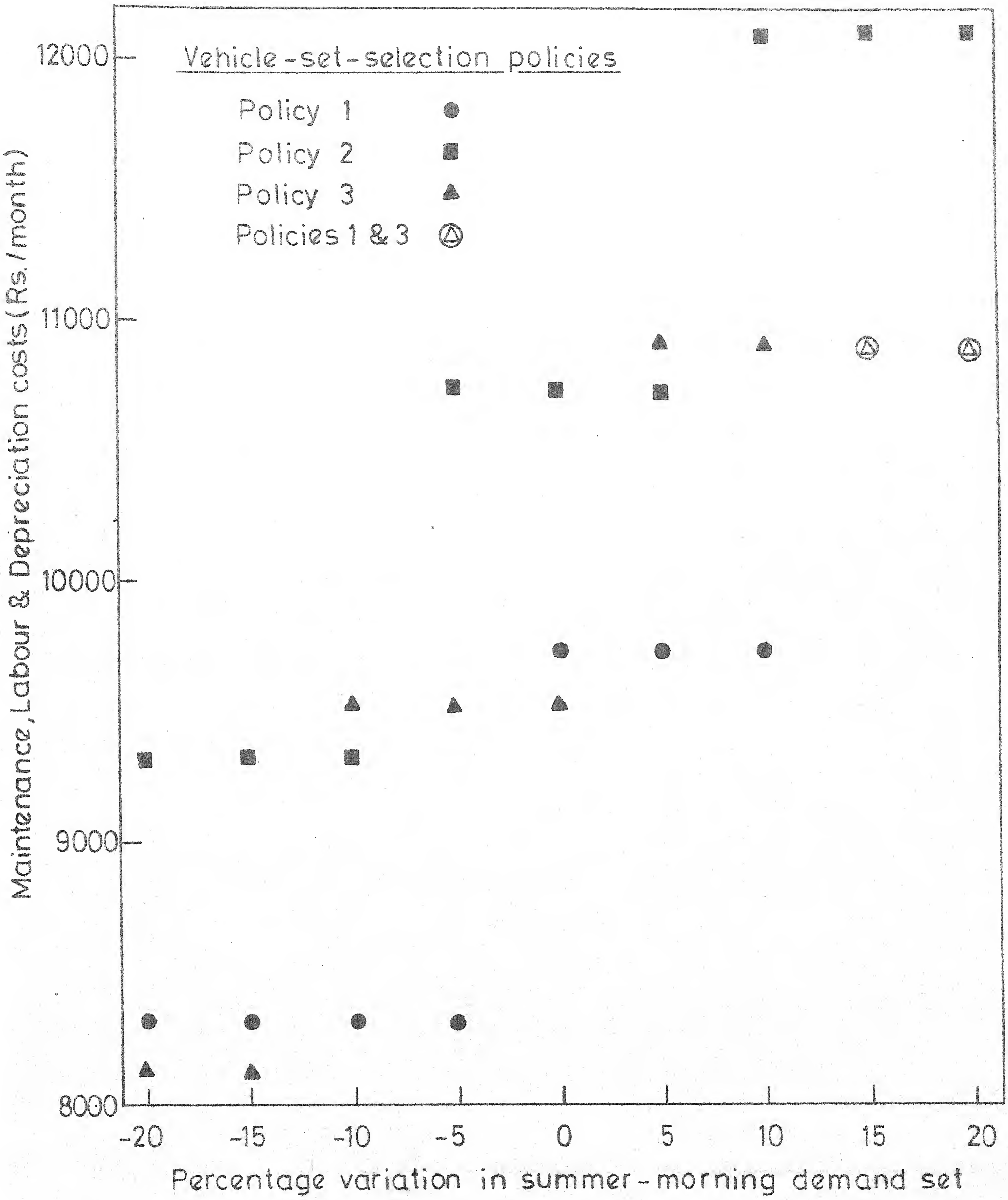

Fig. 1 - Effect of various Vehicle-Set-Selection Policies on the Maintenance, Labour and Depreciation cost, for the different sets of Demands.

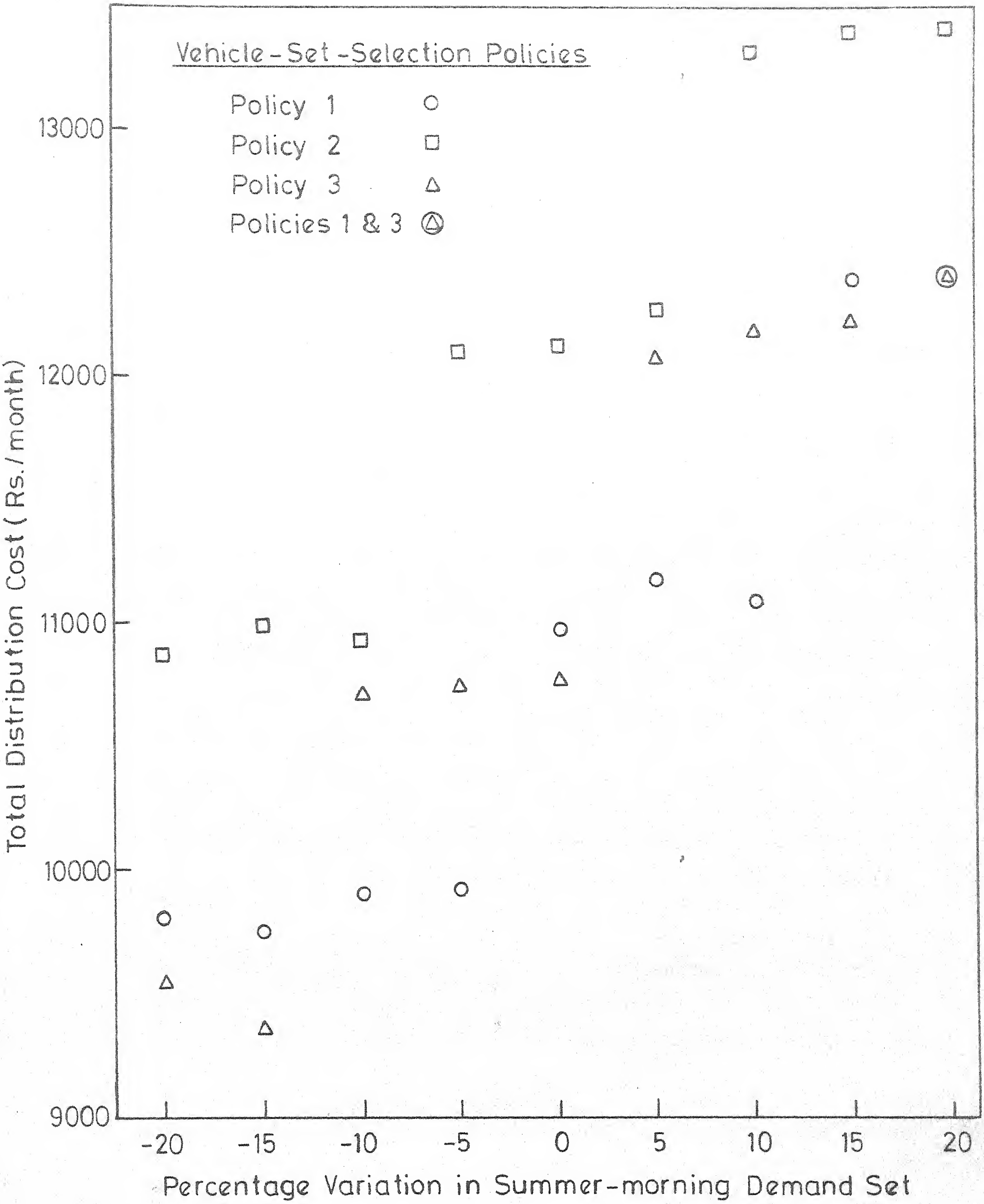

Fig. 2 - Effect of Various Vehicle-Set-Selection Policies on the Total Distribution cost, for the different sets of Demands.

5%, though the maintenance, labour and depreciation costs are observed to be lower in case of policy 2 than in case of policy 3 (Fig. 1), the total distribution cost is observed to be higher with use of policy 2 (as indicated by Fig.2) This is due to the fact that the vehicle-set-selection policy 2 selects more number of small vehicles which use petrol (big vehicles use diesel). Petrol being more expensive than diesel, the cost of transportation is more with use of vehicle-set-selection policy 2. Therefore, one can conclude that the vehicle-set-selection policy 2 for selecting the vehicles is uneconomical. However, as the results show (Fig. 2) there is nothing much to choose between policies 1 & 3. On the other hand, policy 3 being that of random selection, it is recommended that policy 1 be used for implementation.

7.3.2 Effect of Vehicle-allocation policies

Vehicle-allocation policy allocates the vehicles (selected by vehicle-set-selection policies) to the initial routes and then routes are adjusted, accordingly, for the capacity of vehicle allocated to it. Figs. 3 through 5 show the effect of various vehicle allocation policies on the total distribution cost with variation in demand for the particular vehicle-set-selection policy. When the vehicles selected by a vehicle-set-selection policy, are of the same capacities, every vehicle-allocation policy gives the same results as indicated by Fig. 3. This is because the vehicle allocation to the initial routes is the same in every case. However, the various policies yield different results when vehicles of different capacities are employed. No conclusion could be drawn about the superiority of one vehicle allocation policy over the other. Results show (Fig. 3) that there is nothing much to choose between vehicle-allocation policies 1, 2 and 3. Fig. 6 shows the relationship between cost of transportation and demand with use of various vehicle allocation policies.

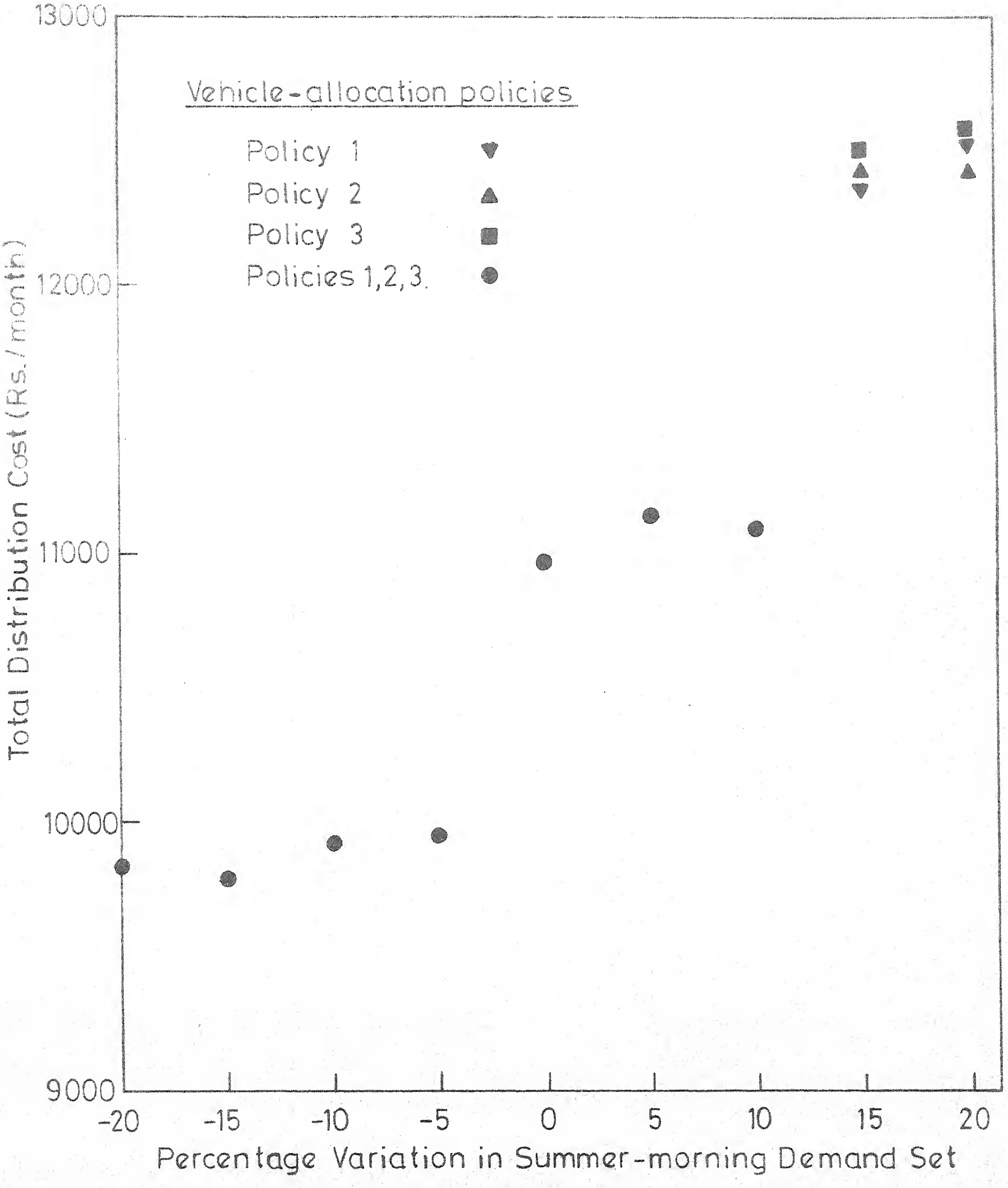

Fig. 3 - Relationship between the Demand and Total Distribution Cost, with use of different Vehicle-allocation Policies and considering Vehicle-Set-Selection Policy 1.

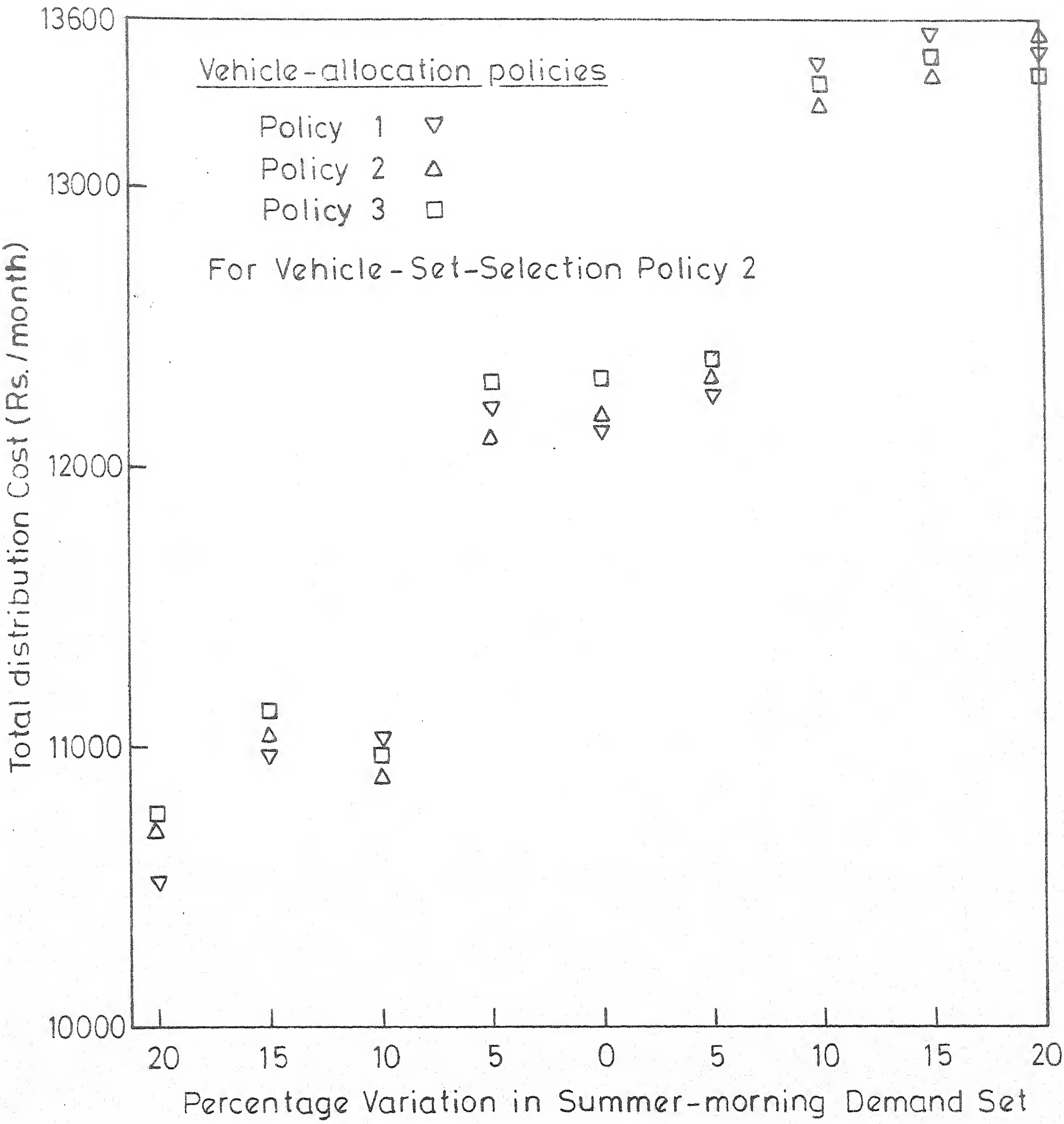

Fig. 4 -Relationship between Demand and Total Distribution Cost, with use of different Vehicle-Alloction Policies and considering Vehicle-Set-Selection Policy 2.

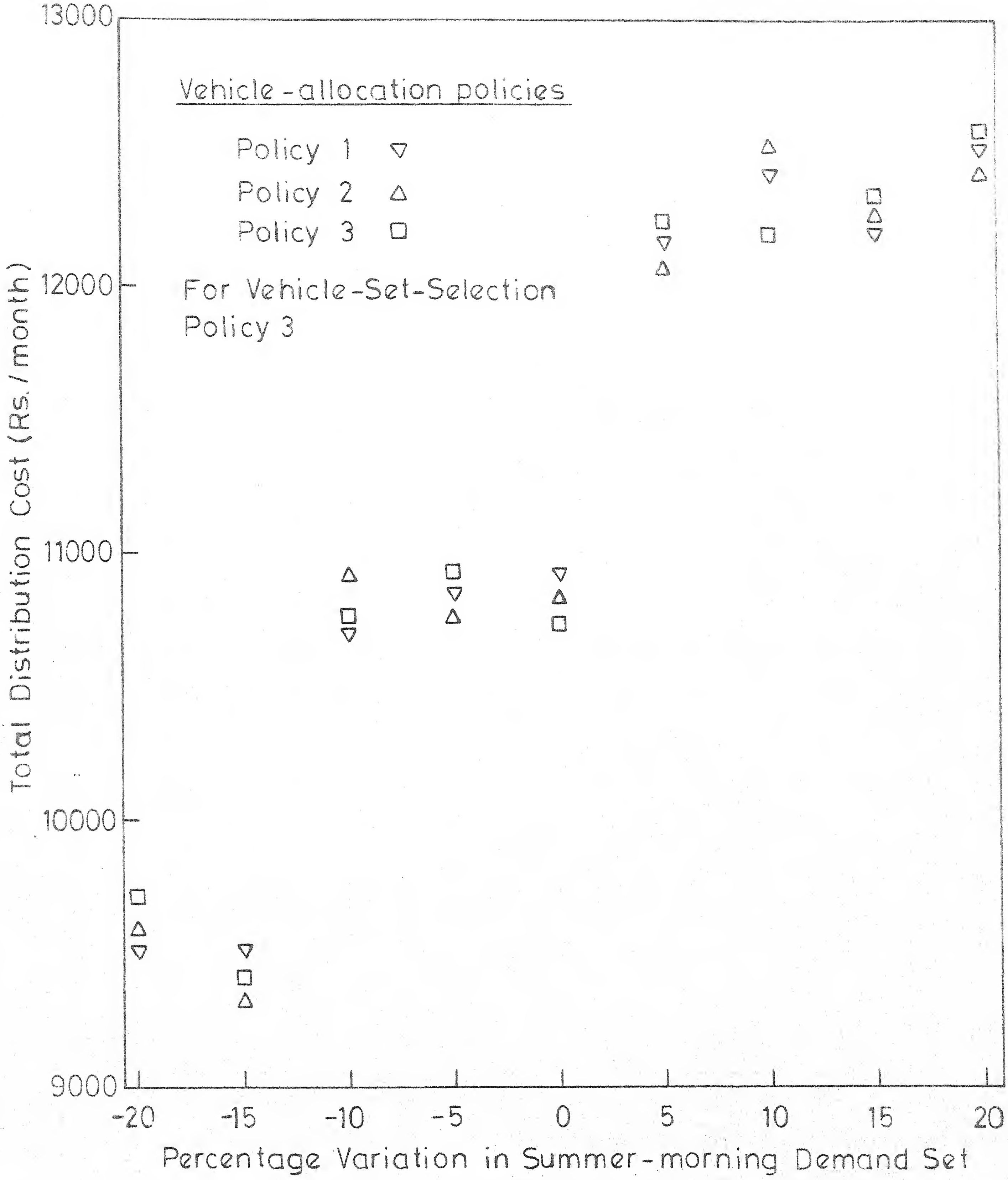

Fig. 5 -Relationship between the Demand and the Total Distribution Cost with use of different Vehicle-Allocation Policies and Considering Vehicle-Set-Selection Policy 2.

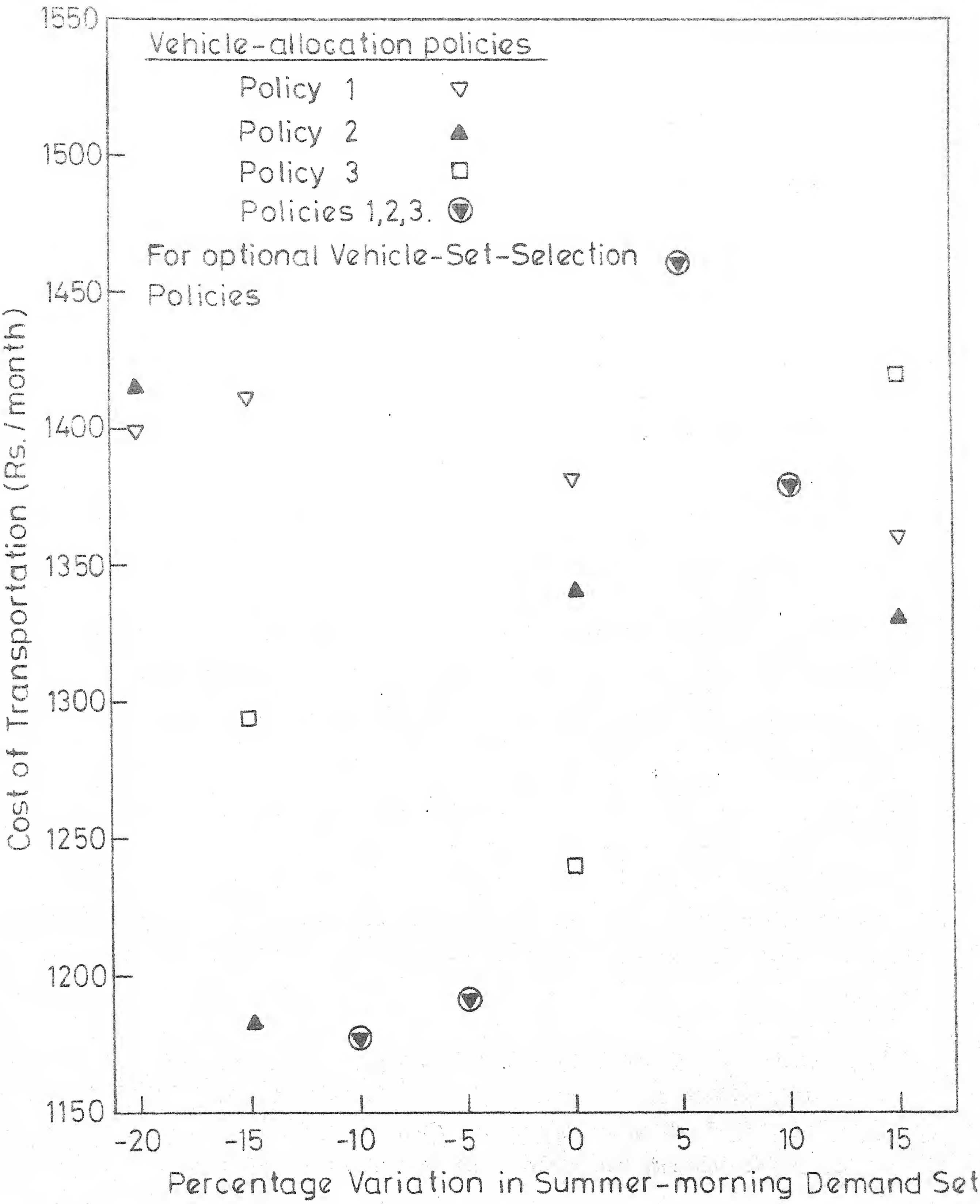

Fig. 6 -Relationship between the Demand and the Cost of Transportation with use of various Vehicle-Allocation Policies.

7.4 Sensitivity Analysis

It consists of changing the values of parameters either one at a time or simultaneously. Here-in, changes have been made only one at a time. Among the various parameters, the parameter subjected to sensitive analysis, in present work, is 'demand' parameter for observing its effect on the objective function. This is due to the fact that demand varies frequently, depending upon the customers' requirement. Table 7.7 and Fig. 7 show the effect of variation in Winter-Morning demand set on the optimal cost of transportation and optimal total distribution cost obtained by the proposed method. It is observed that the total cost of distribution for demand variation of -20% to -6% does not change significantly, maximum being Rs. 8173/- per month and minimum being Rs. 7983/- per month. Similarly, no significant variation in total distribution cost for the present demand set, is observed till the demand at each customer increases by 12%, maximum being Rs. 9547/- p.m. and minimum being Rs. 9494/- p.m.. Because of practical difficulties involved in frequent change of the vehicle dispatching strategies with the change in customers' demand, it is recommended that upto 12% increase in demand at each customer, same vehicle routes and vehicle dispatching strategies are followed, because of no significant change in the optimal cost of distribution. Because of capacity constraints of vehicles, the routes for distribution in practice should be those obtained for maximum demand at each customer i.e., for the demands increased by 12% at each customer. Table 7.8 shows the routes and vehicles allocated to them which can be used for satisfying the customers' demands anywhere between present demands and demands increased by 12% at each customer.

Similarly, Fig. 7 shows that the total cost of distribution, for demand variation between -20% and -6%, does not change significantly, maximum being Rs. 8173/- per month and minimum being Rs. 7983/- per month. Table 7.9 shows the optimal routes and vehicle dispatching strategies

TABLE 7.7

EFFECT OF VARIATION IN WINTER-MORNING DEMAND SET ON OPTIMAL COST OF TRANSPORTATION AND TOTAL COST OF DISTRIBUTION

Percentage variation	Cost of Transportation Rs./month	Total cost of distribution (Rs./month)
-20%	1422.70	8173.70
-15%	1095.50	8040.50
-10%	1038.80	7983.80
-7%	1113.75	8058.75
-6%	1103.60	8048.60
-5%	1303.70	8443.70
0%	1407.00	9547.00
5%	1162.35	9496.35
10%	1184.62	9518.62
12%	1150.20	9494.20
15%	1246.20	10775.20
20%	1368.90	10897.90

Table 7.8

Recommended routes and vehicle-dispatching strategies for the variation of demands at each customer between -20% and -6% of present winter-morning demand set.

Route	Vehicle-allocation Number	Vehicle-allocation Capacity	Route path
1	1	4000	1-15-16-18-21-20-19-1
2	2	4000	1-12-8-17-24-1
3	4	4000	1-10-9-22-23-27-1
4	6	4000	1-26-7-2-4-5-28-1
5	7	4000	1-25-3-6-11-14-13-1

Total cost of distribution = Rs. 8048.6 per month

Table 7.9

Recommended routes and vehicle-dispatching strategies for the variation of demands at each customer upto +12% of present winter-morning demand set

Route	Vehicle-allocation Number	Vehicle-allocation Capacity	Route path
1	1	4000	1-9-3-4-6-11-1
2	2	4000	1-28-22-23-27-1
3	4	4000	1-10-8-2-7-26-1
4	6	4000	1-21-20-18-17-1
5	7	4000	1-25-5-19-16-1
6	8	4000	1-24-15-14-13-12-1

Total cost of distribution = Rs. 9494.2 per month

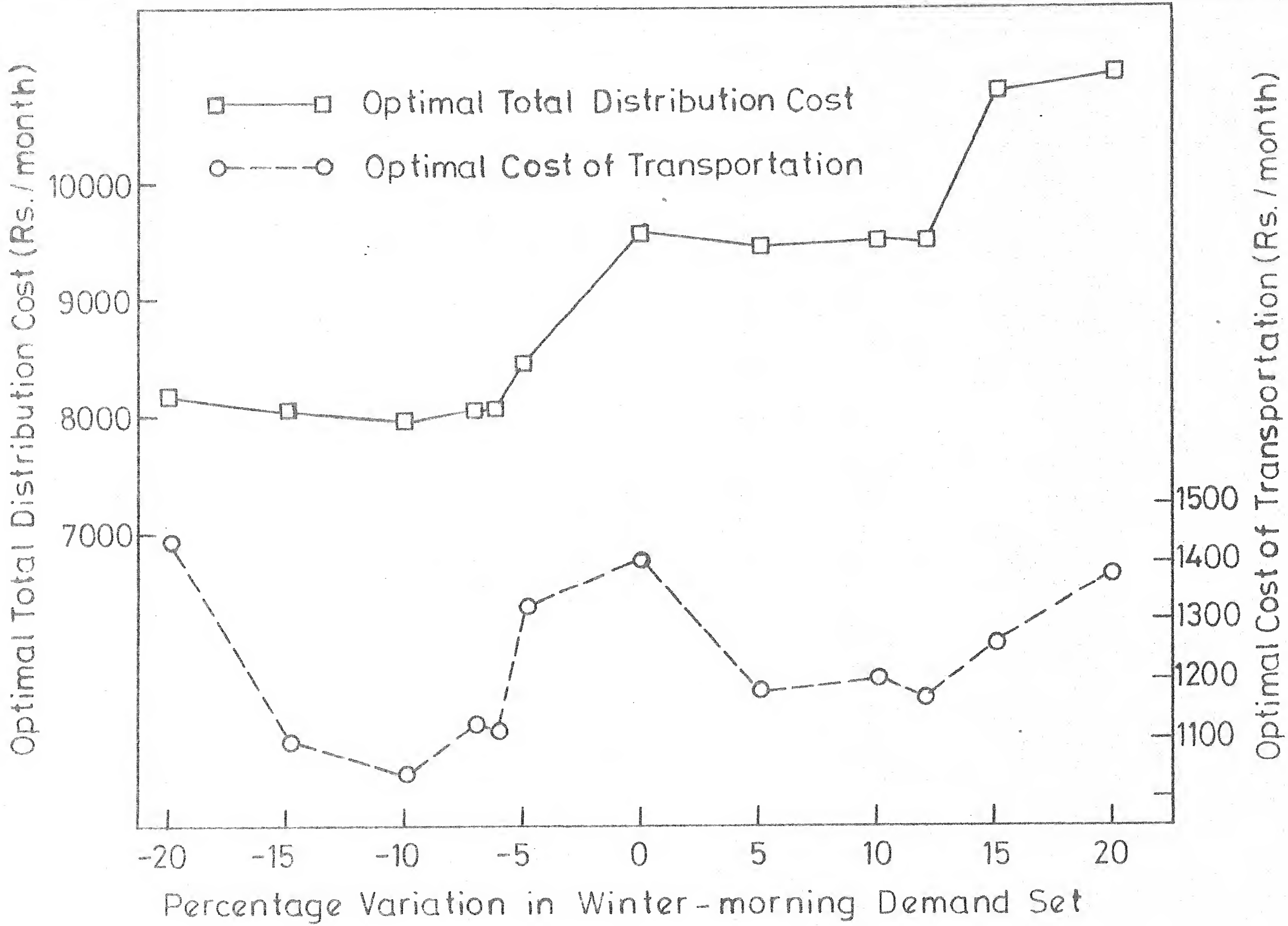

Fig. 7 - Effect of Variation in Winter-morning Demand on Optimal Cost of Transportation and Optimal Distribution Cost.

TABLE 7.10

EFFECT OF VARIATION IN SUMMER MORNING DEMAND SET ON OPTIMAL COST OF TRANSPORTATION AND TOTAL COST OF DISTRIBUTION

Percentage Variation	Cost of transportation (Rs./month)	Total cost of distribution (Rs./month)
-20%	1122	9262
-15%	1181	9321
-10%	1172	9506
-5%	1182	9516
0%	1239	10768
5%	1294	11016
10%	1320	11043
15%	1330	12248
20%	1360	12472

Table 7.11

Recommended routes and vehicle-dispatching strategies for the variation of demands at each customer upto +10% of present summer-morning demand set

Route	Vehicle-allocation No.	Vehicle-allocation Capacity	Route path
1	1	4000	1-5-27-25-1
2	2	4000	1-2-7-11-24-1
3	4	4000	1-10-17-18-16-1
4	6	4000	1-21-20-15-14-13-1
5	7	4000	1-9-28-3-26-1
6	8	4000	1-23-22-19-1
7	9	4000	1-4-6-8-12-1

Total cost of distribution = Rs. 11043.30 per month

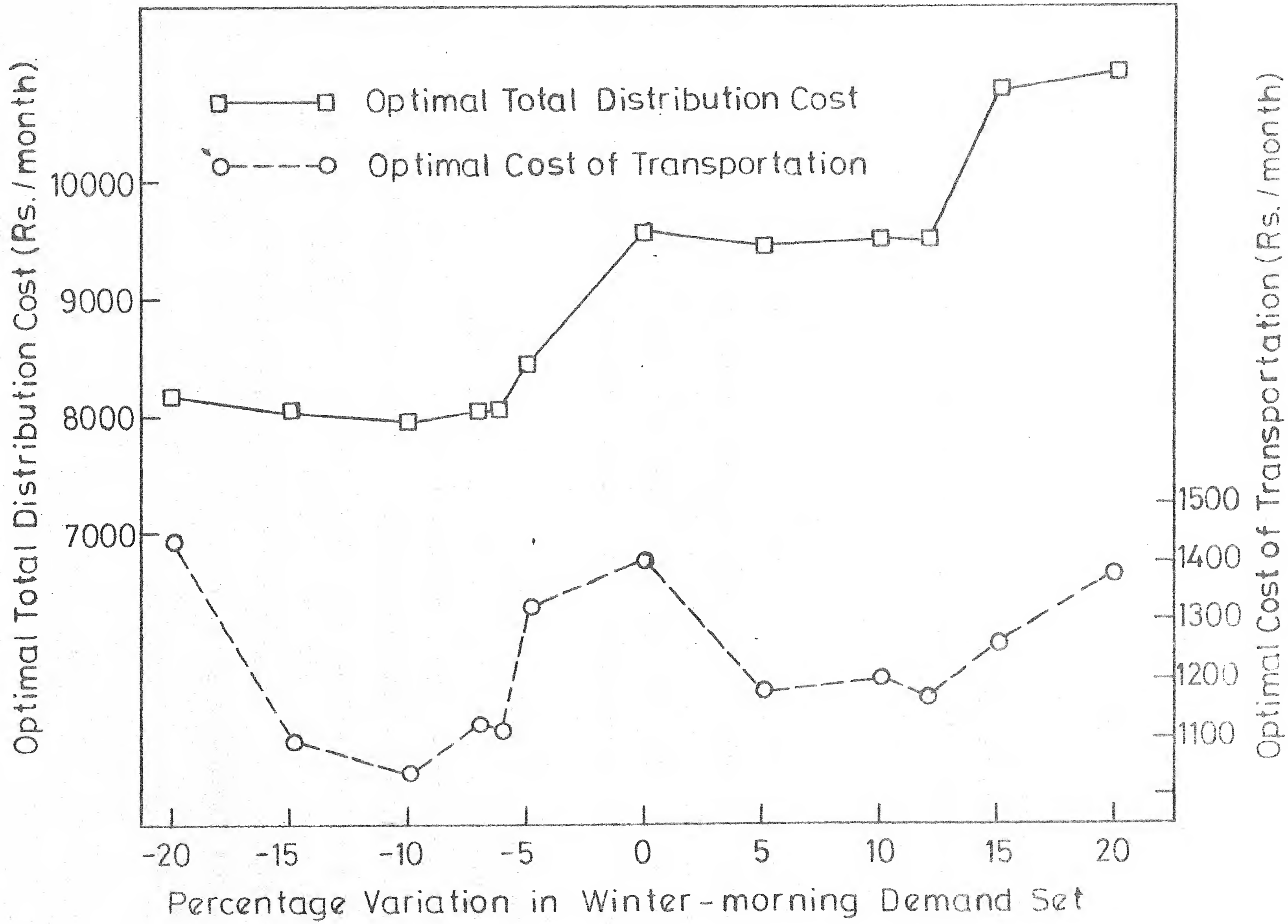

Fig. 7 - Effect of Variation in Winter-morning Demand on Optimal Cost of Transportation and Optimal Distribution Cost.

which are recommended for the variation in demands at each customer anywhere between -6% and -20%. Total cost of distribution, for these recommended routes and policies, is Rs. 8048/- per month, whereas the optimal cost of distribution, for the demands reduced by 10%, is Rs. 7983/- per month. But this difference is not very significant. Therefore, to avoid frequent changes in vehicle dispatching strategies with change in demand, such vehicle dispatching strategies are recommended for the practical purposes. Similarly, the results have been obtained for the variation in Summer-Morning demand set as indicated by Tables 7.10, Table 7.11 and Fig. 8.

7.5 Scope for further work

The work outlined in the previous pages attempts at developing methodology for solving single depot and single product delivery vehicle dispatching problems. For more real life practical problems, the cases of multi-depot and multi-product delivery should be included to develop a general model.

In the problem, as formulated above, it was assumed that at every delivery point the demand must be satisfied by one delivery. As a further extension of the proposed model, this condition, may be relaxed by permitting several routes to pass through a delivery point and partial delivery at the delivery point by the vehicles. Proposed model does not take into consideration the traffic constraint. This constraint may be taken into account for developing a generalized model.

BIBLIOGRAPHY

1. Balinski, M.L. and Quandt, R.E., "On an Integer Program for a delivery problem", Operations Research, Vol. 12, page 300.

2. Bellmore, M. Bennington, "A maximal utility vehicle constrained tanker scheduling problem", Naval Research Logistic Quarterly, Vol. 15 (1968), pages 403-412.

3. Bock, F., 'Mathematical programming solution of Travelling Salesman Examples Recent Advances in Mathematical Programming by R.L. Graves and P. Wolfe, McGraw-Hill book, New York 1963.

4. Christofides, N., Eilon, S., "An algorithm for the Vehicle-dispatching problem", Operations Research Quarterly, 1969, Vol. 20, No. 3, pages 309-318.

5. Christofides N., Eilon, S., "Expected distances in distribution problems", Operational Research Quarterly, 1969, Vol. 20, No.4, pages 437-443.

6. Clarke, G. and Wright, J.W. 'Scheduling of vehicles from a central depot to a number of delivery points", Operations Research, Vol.12,No.2, pages 568-581.

7. Cassidy, P.J. and Benett, H.S., "TRAMP-A multiple-depot vehicle schedulig system" Operational Research Quarterly, Vol 23, No.2, June 72, pages 151-163.

8. Dantzig G.B. and Ramser, J.H., "The truck dispatching problem", Management Science, Volume 6, No. 1, Page 80. 1959

9. Dhingra, C.M., "Design of distribution routes - A case study", Opsearch, Nov-Dec. 1973

10. Dreyfus, S.E., "An appraisal of some shortest path algorithms" Operations Research, Vol. 17, 1969, pages 395=412

11. Eilon, S., C.D.T. Watson Gandy and Christofides, N., "Distribution management", Mathematical modelling and practical treatment, Chapter 6, Charles Griffin.

12. Floyd, 'Algorithm for shortest path" Comm. ACM, Vol. 5, 1962, page 345

13. Fulkerson, D.R., Dantzig, G.B. and Johnson, S.M., "Solution of a large scale travelling salesman problem", Operations Research, 1954, Vol. 2, page 393-410.

14. Gould, F.J., "A Linear Programming Model for Cargo Movement evaluation", Transportation Science, Vol. 5,1971, No. 4, Pages 344-365

15. Gaskell , T.J. "Bases for Vehicle fleet scheduling", Operational Research Quarterly, 1967, Vol. 18, Page 281

16. Gould, J., "The Size and comparison of a Road transport fleet", Operational Research Quarterly, 1969, Vol. 20, No. 1, pages 81-92.

17. Hu, T.C., "Integer Programming and Network Flows", Addison, Wesley, Mass, 1969

18. Hayes, R.L., "The Delivery Problem", Presented at the thirty-first national meeting of Operations Research Society of America, New York, N.Y., May 31-June 2, 1967.

19. Joseph, A. Svestka and Vaughn E. Huckfeldt, "Computational experience with an M-Salesman travelling salesman algorith", Management Science, March 1973, Page 790.

20. John E. Biegel, "Production Control", Prentice Hall, INC. New Jersey, 1971.

21. James L. Riggs, "Production Systems", John Wiley and Sons, Inc., New Jersey, 1970.

22. Keith V. Smith, Charles T. Phillips and Ralph J. Lewis, "Network evaluation of complex transportation system", Transportation Research, Vol. 6, 1972, No. 1, pages 103-111.

23. Lin, S. and Kernighan, B.W., "An effective heuristic algorithm for the travelling salesman problem" Operations Research Vol. 21, No. 2, March-April 1973, Page 498.

24. Little, J.D.C., Murty, K.G. and Sweeney D.W., "An algorithm for the travelling salesman problem", Operations Research, Vol. 11,1963, Page 972.

25. Loubal, P.S., "A travelling salesman algorithm suitable for vehicle routing problems" - Interdepartmental Working Paper No. 3, Institituion of Transportation and Traffic Engineering, University of California, Berkeley, July 1964.

26. Mann, A.F., "Implementation of Computer Vehicle scheduling computer programme". Paper presented to the North Western Operational Research Group Conferences, 1966.

27. Miller, Tucker and Zemlin, "Integer Programming formulation of travelling salesman problems", J.ACM , Vol. 7, 1960, Pages 326-329.

28. Norman, W.E., "The impact of computer techniques on Road Transport Planning", National Computing Centre, Manchester, 1969.

29. ____________ "Computers in vehicle scheduling", National Computing Centre, Manchester, 1969.

30. Nicholson, "Finding shortest route between two points in a network", Computer Journal, Vol. 9, No. 3, Pages 275-280.

31. Pandit, S.N.N., "Some observations on the Routing Problem" Operations Research, Vol. 10, 1962, Page 726-727.

32. Tyagi, M.S.. "A practical method for truck dispatching problem", Journal of Operations Research Society of Japan, Vol. 10, 1968, No. 3 and 4, pages 76-96.

33. Watson, C.D.T. and Eilon, S., "The depot siting problem with discontinuous delivery cost", Operational Research Quarterly, Vol. 23, No. 3, Sept. 1972, Page 277.

34. Wren, A., Holliday, A., "Computer scheduling of vehicles from one or more depots to a number of delivery points". Operational Research Quarterly, Vol. 23, No. 3, Sept. 1972, Page 333.

35. Wren, A., "Computers in Transport Planning and Operation",1971
Ian Allan Shepperton.

36. Webb, M.J., "Vehicle routing and scheduling by computer", Paper
presented to the Operational Research Conference,
Shrivenham, 1965.

* APPENDIX

```
*************************************************************
*                                                           *
*            COMPUTER  LISTING                              *
*                                                           *
*************************************************************
```

```
*************************************************************
*                                                           *
*  DETERMINATION OF OPTIMAL ROUTES AND VEHICLE              *
*  DISPATCHING STRATEGIES FOR A DISTRIBUTION NETWORK        *
*                                                           *
*************************************************************
```

```
      ************************************************************
      *     PROGRAM  1  FOR  CONSTRUCTING INITIAL  ROUTES      *
      ************************************************************
      DIMENSION KDD(40,40),MINSEQ(20,40),Z(50),NEQ(20),LINSEQ(20,40)      PRO0003
      DIMENSION ID(40,40),MMTP(40),LEQ(20),IES(20),LIN(50),LES(20)        PRO0004
      DIMENSION SPTS(40),SPPS(40),DD(40,40),JCB1(40),JCB2(40)
      DIMENSION KINSEQ(20,40),KKEQ(20)                                    PRO0006
      DIMENSION KDSD(32,32)                                               PRO0006
      INTEGER Z,ZZ,ZLB,ZUB,DD,SPTS,SPPS
      COMMON/SLM/MINSEQ,KK
      COMMON/RNC/LINSEQ,KDD,N
      COMMON/VSS/DD
      COMMON/NEEN/NV
      READ 6083,A                                                         PRO0011
 6083 FORMAT(F5.1)                                                        PRO0011
 490  READ 1,NC                                                           PRO0012
 1    FORMAT(I3)                                                          PRO0013
      IF(NC.EQ.100)GO TO 9000                                             PRO0013
      NCC=NC+1                                                            PRO0014
      A=A+5.0                                                             PRO0014
      PRINT 6053,NC                                                       PRO0014
 6053 FORMAT(1H1,//10X,*NO. OF CUSTOMERS =*,I4/)                          PRO0014
      PRINT 6054                                                          PRO0014
 6054 FORMAT(//20X,*DISTANCE MATRIX*//)                                   PRO0014
      PRINT 6055,((I),I=1,NCC)                                            PRO0014
 6055 FORMAT(3X,32I4)                                                     PRO0014
      DO 101 I=1,NC                                                       PRO0015
      JJ=I+1                                                              PRO0016
      READ 102,(KDSD(I,II),II=JJ,NCC)                                     PRO0017
 101  CONTINUE                                                            PRO0018
 102  FORMAT(20I4)                                                        PRO0019
      DO 103 I=1,NCC                                                      PRO0019
      DO 103 J=1,NCC                                                      PRO0019
      IF(J.GT.I)GO TO 103                                                 PRO0019
      IF(J.EQ.I)GO TO 104                                                 PRO0019
      KDSD(I,J)=KDSD(J,I)                                                 PRO0019
      GO TO 103                                                           PRO0019
 104  KDSD(I,J)=9999                                                      PRO0019
 103  CONTINUE                                                            PRO0020
      DO 6020 I=1,NCC                                                     PRO0020
      PRINT 6030,I,(KDSD(I,J),J=1,NCC)                                    PRO0020
 6020 CONTINUE                                                            PRO0020
 6030 FORMAT(I3,1X,32I4)                                                  PRO0020
 6000 READ 2,NV                                                           PRO0020
 2    FORMAT(I3)                                                          PRO0020
      IF(NV.EQ.500)GO TO 490                                              PRO0020
      PRINT 6007,NC,NV                                                    PRO0020
 6007 FORMAT(1H1,///9X,*NO. OF CUSTMRS=*,I3/9X,*NO. OF VEHICLE=*,I3/)    PRO0020
      NVV=NV+1                                                            PRO0020
      N=NV+NC                                                             PRO0020
      DO 6070 I=1,NV                                                      PRO0020
      DO 6070 J=1,N                                                       PRO0020
      IF(J.GT.NV)GO TO 6080                                               PRO0020
      KDD(I,J)=9999                                                       PRO0020
      GO TO 6070                                                          PRO0020
 6080 JJ=J-NV+1                                                           PRO0020
      KDD(I,J)=KDSD(1,JJ)                                                 PRO0020
 6070 CONTINUE                                                            PRO0020
      PRINT 5050                                                          PRO0020
```

```
5050 FORMAT(//10X,*INITIAL ROUTES BY MULTIPLE TRAVELLING SALESMEN PRO*)
     DO 6090 I=NVV,N
     II=I-NV+1
     DO 6090 J=1,N
     IF(J.GT.NV)GO TO 6095
     KDD(I,J)=KDSD(II,1)
     GO TO 6090
6095 JJ=J-NV+1
     KDD(I,J)=KDSD(II,JJ)
6090 CONTINUE
     IF(NV.EQ.2)GO TO 6081
     GO TO 6088
6081 PRINT 740,((I),I=1,N)
740  FORMAT(//15X,*AUGMENTED DISTANCE MATRIX*,//4X,40(I3))
     DO 730 I=1,N
     PRINT 720,I,(KDD(I,J),J=1,N)
730  CONTINUE
720  FORMAT(I3,2X,40I3)
6088 DO 5 I=1,N
     DO 5 J=1,N
     DD(I,J)=KDD(I,J)
  5  CONTINUE
     CALL ASGNM(MINSEQ,ZZ,KN,NEQ,N)
     IF(KN.EQ.1000)GO TO 9008
     PRINT 710
710  FORMAT(//15X,*INITIAL ASSIGNMENT GIVES FOLLOWING SUBTOURS*,/)
     PRINT 6079
6079 FORMAT(/1X,*SUBTOUR*,2X,*PATH SEQUENCE*,/)
     DO 7 I=1,KN
     KPS=NEQ(I)
     PRINT 8,I,(MINSEQ(I,J),J=1,KPS)
8    FORMAT(1X,I3,5X,41(I3))
  7  CONTINUE
     PRINT 9,ZZ
  9  FORMAT(/10X,* INITIAL LOWER BOUND=*,I5)
     CALL FEAS(MINSEQ,NEQ,KN,IES,KK)
     IF(KK.EQ.0)GO TO 500
C*****FINDING INITIAL UPPER BOUND
     KNZ=0
     DO 10 I=1,KN
     KPSP=NEQ(I)
     DO 11 J=1,KPSP
     LINSEQ(I,J)=MINSEQ(I,J)
 11  CONTINUE
     LEQ(I)=KPSP
 10  CONTINUE
     LL=KK
     DO 12 I=1,KK
     LES(I)=IES(I)
 12  CONTINUE
     KMN=KN
     CALL UPRBND(KZUB,LEQ,LES,LL,KMN)
     ZUB=KZUB
     PRINT 810,ZUB
810  FORMAT(//10X,*BELOW TOURS BY INITIAL HEURISTICS*,10X,*U.B.=*I5/)
     DO 811 I=1,KMN
     JOIT=LEQ(I)
     PRINT 812,(LINSEQ(I,J),J=1,JOIT)
812  FORMAT(5X,41(I3))
```

```
811   CONTINUE
      IF(ZUB.EQ.ZZ)GO TO 600
      GO TO 650
500   PRINT 13
 13   FORMAT(/5X,*ABOVE TOURS ARE FINAL AND MIN DIS IS INITIAL L.B.*)
      GO TO 6000                                                       PRO0083
600   PRINT  16
 16   FORMAT(//10X,*FINAL TOURS ARE AS OBTAINED BY INITIAL HEURISTICS*) PRO0084
      GO TO 6000                                                       PRO0085
650   DO 663 I=1,N
      DO 663 J=1,N
      ID(I,J)=KDD(I,J)
663   CONTINUE
 700 CALL SHORT(IES,NEQ,MMTP,LMS,LIN)
      LNS=LMS-1
      MM=0
      DO 120 I=1,LNS
      II=I+1
      MM=MM+1
      SPTS(MM)=MMTP(I)
      SPPS(MM)=MMTP(II)
120   CONTINUE
      KONT=0                                                           PRO0101
      DO 130 JHUM=1,MM
      DO 131 I=1,N
      DO 131 J=1,N
      DD(I,J)=ID(I,J)
131   CONTINUE
      KTE1=SPTS(JHUM)
      KTE2=SPPS(JHUM)
      DD(KTE1,KTE2)=9999
      CALL ASGNM(MINSEQ,ZZ,KN,NEQ,N)
      IF(KN.EQ.1000)GO TO 9008
      IF(ZZ.GE.ZUB) GO TO 135
      KONT=KONT+1
      Z(KONT)=ZZ
      JOB1(KONT)=KTE1
      JOB2(KONT)=KTE2
      CALL FEAS(MINSEQ,NEQ,KN,IES,KK)
      IF(KK.EQ.0)GO TO 140
      GO TO 130
140   ZUB=ZZ
      KNZ=KN
      DO 7054 I=1,KNZ
7054  KKEQ(I)=NEQ(I)
      DO 141 I=1,KNZ
      KPSE=KKEQ(I)
      DO 746 J=1,KPSE
      KINSEQ(I,J)=MINSEQ(I,J)
746   CONTINUE
142   FORMAT(1X,I3,5X,41I3)                                            PRO014
141   CONTINUE
      GO TO 130
135   CONTINUE                                                         PRO014
130   CONTINUE
      IF(KONT.EQ.0)GO TO 2000
      CALL MIN(Z,KONT,KENT,MKONT)
      ZLB=MKONT
      IF(ZLB.EQ.ZUB)GO TO 2000
```

```
      KTE3=JOB1(KENT)
      KTE4=JOB2(KENT)
      DO 695 I=1,N
      DO 695 J=1,N
      DD(I,J)=ID(I,J)
695   CONTINUE
      DD(KTE3,KTE4)=9999
      CALL ASGNM(MINSEQ,ZZ,KN,NEQ,N)
      IF(KN.EQ.1000)GO TO 9008
      DO 691 I=1,N
      DO 691 J=1,N
      ID(I,J)=DD(I,J)
691   CONTINUE
      CALL FEAS(MINSEQ,NEQ,KN,IES,KK)
      IF(KK.EQ.0)GO TO 3000
      GO TO 700
3000  PRINT 3003,ZZ
3003  FORMAT(/10X,*FINAL TOURS ARE AS FOLLOWS*,5X,*ZZ=*I5)
      PRINT 6079                                                      PRO01
      DO 3001 I=1,KN
      KPZS=NEQ(I)
      PRINT 3002,I,(MINSEQ(I,J),J=1,KPZS)
3002  FORMAT(1X,I3,5X,41(I3))                                         PRO01
3001  CONTINUE
      GO TO 6000                                                      PRO01
2000  IF(KNZ.EQ.0)GO TO 7808                                          PRO01
      PRINT 2001,ZUB                                                  PRO01
2001  FORMAT(//20X,*FOLLOWING TOURS ARE OPTIMAL*,10X,*MIN DIS =*,I5)
      PRINT 6079                                                      PRO01
      DO 2050 I=1,KNZ
      KPSE=KKEQ(I)
      PRINT 2060,I,(KINSEQ(I,J),J=1,KPSE)                             PRO01
2060  FORMAT(1X,I3,5X,41(I3))                                         PRO01
2050  CONTINUE
      GO TO 6000                                                      PRO02
7808  PRINT 16                                                        PRO02
      GO TO 6000                                                      PRO02
9008  PRINT 9007
9007  FORMAT(//1X,*MISTAKE IN FINDING MIN.LIN.REQD.CHECK SUBR.*)
      GO TO 6000                                                      PRO02
9000  STOP
      END
```

```
 ISN        SJURCE STATEMENT

   0 $IBFTC FEAS
   1          SJBROUTINE FEAS(NINSEQ,KEQ,KLN,KES,MK)
   2          CJMMON/NEEN/NV
   3          DIMENSION NINSEQ(20,40),KEQ(20),KES(20)
   4          MK=0
   5          DJ 5 I=1,KLN
   6          KPS=KEQ(I)
   7          DJ 6 J=1,KPS
  10          KTT=NINSEQ(I,J)
  11          DJ 7 JJ=1,NV
  12          IF(KTT.EQ.JJ)GJ TO 5
  15     7    CJNTINUE
  17     6    CJNTINUE
  21          MK=MK+1
  22          KES(MK)=I
  23     5    CJNTINUE
  25          RETURN
  26          END
```


NO MESSAGES FJR ABOVE ASSEMBLY

```
   ISN        SOURCE STATEMENT

     0  $IBFTC SHORT                                                              PR
     1         SUBROUTINE SHORT(IES,NEQ,MTP,IMS,KIN)                              PR
     2         DIMENSION IES(20),NEQ(20),MTP(40),KIN(50),MINSEQ(20,40)            PR
     3         COMMON/SLM/MINSEQ,KK                                               PR
     4         KT=0                                                               PR
     5         DO 11 I=1,KK                                                       PR
     6         INS=IES(I)                                                         PR
     7         KIN(I)=NEQ(INS)                                                    PR
    10    11   CONTINUE                                                           PR
    12         CALL MIN(KIN,KK,KT,MF)                                             PR
    13         JTP=IES(KT)                                                        PR
    14         IMS=NEQ(JTP)                                                       PR
    15         DO 2 J=1,IMS                                                       PR
    16         MTP(J)=MINSEQ(JTP,J)                                               PR
    17     2   CONTINUE                                                           PR
    21         RETURN                                                             PR
    22         END                                                                PR
```

1EG250   IBMAP ASSEMBLY SHORT

NO MESSAGES FOR ABOVE ASSEMBLY

```
EG250                                      FORTRAN SOURCE LIST
   ISN        SOURCE STATEMENT

     0 $IBFTC UPRBND
     1         SUBROUTINE UPRBND(KZUB,LEQ,LES,LL,KMN)
     2         COMMON/RUM/IP
     3         DIMENSION LINSEQ(20,40),KDD(40,40),LEQ(20),LES(20),IP(40,40)
     4         DIMENSION IFEAS(20),INFEAS(20)
     5         COMMON/RNC/LINSEQ,KDD,N
     6         DO 1 I=1,N
     7         DO 1 J=1,N
    10         IP(I,J)=0
    11     1   CONTINUE
    14         CALL PMATRX(KMN,LEQ)
    15   100   L=0
    16         K=0
    17         DO 2 I=1,KMN
    20         DO 3 J=1,LL
    21         IF(I.EQ.LES(J))GO TO 4
    24     3   CONTINUE
    26         K=K+1
    27         IFEAS(K)=I
    30         GO TO 2
    31     4   L=L+1
    32         INFEAS(L)=I
    33     2   CONTINUE
    35         CALL SEARCH(L,K,INFEAS,IFEAS,LEQ,N1,N2,N3,N4,IFFST,IBERT)
    36         IP(N1,N3)=1
    37         IP(N1,N2)=2
    40         IP(N4,N2)=1
    41         IP(N4,N3)=2
    42         CALL MNSQ(IP,KDD,N,LINSEQ,LEQ,KMN,IZUB)
    43         CALL FEAS(LINSEQ,LEQ,KMN,LES,LL)
    44         IF(LL.EQ.0)GO TO 1005
    47         GO TO 100
    50  1005   KZUB=IZUB
    51         RETURN
    52         END
MEG250                                     IBMAP ASSEMBLY UPRBND


  NO MESSAGES FOR ABOVE ASSEMBLY
```

```
ISN       SOURCE STATEMENT

  0 $IBFTC PMATRX
  1         SUBROUTINE PMATRX(KMN,LEQ)
  2         DIMENSION LINSEQ(20,40),IP(40,40),LEQ(20),KDD(40,40)
  3         COMMON/RNC/LINSEQ,KDD,N
  4         COMMON/RUM/IP
  5         DO 1 I=1,KMN
  6         KPS=LEQ(I)
  7         KPN=KPS-1
 10         DO 2 J=1,KPN
 11         KTT=LINSEQ(I,J)
 12         JJ=J+1
 13         KTM=LINSEQ(I,JJ)
 14         IP(KTT,KTM)=1
 15         DO 3 K=1,N
 16         IF(K.EQ.KTM)GO TO 3
 21         IP(KTT,K)=2
 22   3     CONTINUE
 24         DO 4 KK=1,N
 25         IF(KK.EQ.KTT)GO TO 4
 30         IP(KK,KTM)=2
 31   4     CONTINUE
 33   2     CONTINUE
 35   1     CONTINUE
 37         RETURN
 40         END
```


NO MESSAGES FOR ABOVE ASSEMBLY

```
.G250                                              FORTRAN SOURCE LIST
   ISN       SJURCE STATEMENT

     0  $IBFTC MNSQ
     1          SJBRJUTINE MNSQ(P,MD,N,MANSEQ,NEQ,KN,IZ)
     2          DIMENSION P(40,40),MD(40,40),MANSEQ(20,40),NEQ(20)
     3          INTEGER P
     4          DIMENSION JINSEQ(100)
     5          N1=2*N
     6          KN=0
     7          DJ 401 K=1,N1
    10          JINSEQ(K)=0
    11    401   CJNTINUE
    13          KJM=0
    14          KK=0
    15          IZ=0
    16          DJ 402 I=1,N
    17          DJ 403 K=1,N1
    20          IF(I.EQ.JINSEQ(K))GO TO 402
    23    403   CJNTINUE
    25          KK=KK+1
    26          JINSEQ(KK)=I
    27          DJ 404 J=1,N
    30          J1=J
    31          IF(P(I,J).EQ.1)GO TO 405
    34    404   CJNTINUE
    36    405   KK=KK+1
    37          JINSEQ(KK)=J1
    40          IZ=IZ+MD(I,J1)
    41    406   J2=JINSEQ(KK)
    42          DJ 407 J=1,N
    43          IF(P(J2,J).EQ.1)GO TO 410
    46    407   CJNTINUE
    50    410   KK=KK+1
    51          JINSEQ(KK)=J
    52          IZ=IZ+MD(J2,J)
    53          IF(J.EQ.I)GO TO 431
    56          GJ TO 406
    57    431   KN=KN+1
    60          KIM=KK-KJM
    61          KJM=KK
    62          NEQ(KN)=KIM
    63          KZ=KK-KN
    64          IF(KZ.GE.N)GO TO 590
    67    402   CJNTINUE
    71    590   JSS=0
    72          DJ 1001 I=1,KN
    73          NERC=NEQ(I)
    74          DJ 1003 J=1,NERO
    75          K=J+JSS
    76          MANSEQ(I,J)=JINSEQ(K)
    77    1003  CJNTINUE
   101          JSS=K
   102    1001  CJNTINUE
   104          RETURN
   105          END
MEG250                                             IBMAP ASSEMBLY MNSQ

  NO MESSAGES FJR ABOVE ASSEMBLY
```

```
ISN        SOURCE STATEMENT

   $IBFTC SEARCH
         SUBROUTINE SEARCH(L,K,INFEAS,IFEAS,NEQ,N1,N2,N3,N4,IFFST,IFFRT)
         COMMON/RNC/LINSEQ,KDD,N
         DIMENSION LINSEQ(20,40),KDD(40,40),INFEAS(20),IFEAS(20),NEQ(20)
         DIMENSION IXP(40),KIXP(40),IPX(40),KIPX(40),IFST(20),IFRT(20)
         DIMENSION IIPX(40),KKIPX(40),I1(20),I2(20),I3(20),I4(20)
         DIMENSION NDIS(50),MINDIS(50),MINMUM(50),IFRT(20),MININC(20)
         DIMENSION J1(20),J2(20),J3(20),J4(20)
         KZ=0
         DO 3 I=1,L
         IM=INFEAS(I)
         KPS=NEQ(IM)
         KW=0
         DO 4 J=1,K
         IN=IFEAS(J)
         KPN=NEQ(IN)
         KRN=KPN-1
         KV=0
         DO 5 JJ=1,KRN
         KV=KV+1
         KTT=LINSEQ(IN,JJ)
         JK=JJ+1
         KTM=LINSEQ(IN,JK)
         IXP(KV)=KTT
         KIXP(KV)=KTM
         KQ=0
         DO 6 II=2,KPS
         KTS=LINSEQ(IM,II)
         IK=II-1
         KTN=LINSEQ(IM,IK)
         IDIS=KDD(KTT,KTS)
         JDIS=KDD(KTN,KTM)
         KDIS=KDD(KTN,KTS)
         KQ=KQ+1
         NDIS(KQ)=IDIS+JDIS-KDIS
         IPX(KQ)=KTS
         KIPX(KQ)=KTN
   6     CONTINUE
         CALL MIN(NDIS,KQ,KQQ,MET)
         IIPX(KV)=IPX(KQQ)
         KKIPX(KV)=KIPX(KQQ)
         MDIS=KDD(KTT,KTM)
         MINDIS(KV)=MET-MDIS
   5     CONTINUE
         CALL MIN(MINDIS,KV,KVV,MER)
         KW=KW+1
         MINMUM(KW)=MER
         I1(KW)=IXP(KVV)
         I2(KW)=KIXP(KVV)
         I3(KW)=IIPX(KVV)
         I4(KW)=KKIPX(KVV)
         IFRT(KW)=IN
   4     CONTINUE
         CALL MIN(MINMUM,KW,KWW,MEP)
         KZ=KZ+1
```

```
ISN      SOURCE STATEMENT

 72         MININC(KZ)=MEP                                         PR
 73         J1(KZ)=I1(KWW)                                         PR
 74         J2(KZ)=I2(KWW)                                         PR
 75         J3(KZ)=I3(KWW)                                         PR
 76         J4(KZ)=I4(KWW)                                         PR
 77         IFST(KZ)=IFRT(KWW)                                     PR
100         IERT(KZ)=IM                                            PR
101   3     CONTINUE                                               PR
103         CALL MIN(MININC,KZ,KZZ,MEM)                            PR
104         MINI=MEM                                               PR
105         N1=J1(KZZ)                                             PR
106         N2=J2(KZZ)                                             PR
107         N3=J3(KZZ)                                             PR
110         N4=J4(KZZ)                                             PR
111         IFFST=IFST(KZZ)                                        PR
112         IEERT=IERT(KZZ)                                        PR
113         RETURN                                                 PR
114         END                                                    PR
```


NO MESSAGES FOR ABOVE ASSEMBLY

ISN SOURCE STATEMENT

```
$IBFTC ASGNM
       SUBROUTINE ASGNM(MANSEQ,IZ,KN,NEQ,N)
       INTEGER DD,P,UNMRKR,UC,UR,UZ,D
       COMMON/VSS/DD
       COMMON/JLM/P
       COMMON/ZLM/D
       DIMENSION DD(40,40),P(40,40),UNMRKR(40),MRRKC(50)
       DIMENSION D(40,40),IPOK(40),ISOK(40),NDIS(100),IXP(100),IPX(100)
       DIMENSION MANSEQ(20,40),NEQ(20)
       DO 710 I=1,N
       DO 710 J=1,N
       D(I,J)=DD(I,J)
  710  CONTINUE
  160  CALL ZERO(N)
       DO 107 I=1,N
       DO 107 J=1,N
       P(I,J)=0
  107  CONTINUE
       CALL ASSIGN(N,IPOK,ISOK,KOK)
  995  INDEX=0
       DO 123 I=1,N
       DO 123 J=1,N
       IF(P(I,J).EQ.1)INDEX=INDEX+1
  123  CONTINUE
       IF(INDEX.EQ.N)GO TO 110
       CALL MINLIN(UNMRKR,UR,MRRKC,UC,N)
       UZ=UR+UC
       IF(INDEX.EQ.UZ)GO TO 150
       IF(INDEX.GT.UZ)GO TO 118
       GO TO 141
  118  PRINT 119
  119  FORMAT(//10X,*SOMETHING WRONG IN FINDING MIN. LINES REQD.*)
       GO TO 5000
  141  IF(UZ.LT.N)GO TO 150
       GO TO 166
  150  CALL CRZERO(N,UNMRKR,MRRKC,UR,UC)
       GO TO 160
  110  CALL MNSQ(P,DD,N,MANSEQ,NEQ,KN,IZ)
       GO TO 500
  166  DO 920 I=1,N
       KAKI=0
       DO 921 J=1,N
       IF(P(I,J).EQ.1)GO TO 920
  921  CONTINUE
       DO 922 K=1,N
       IDIS=DD(I,K)
       DO 923 KK=1,N
       IF(P(KK,K).EQ.1)GO TO 924
  923  CONTINUE
       GO TO 925
  924  DO 950 J=1,N
       IF(P(KK,J).EQ.1)GO TO 950
       DO 990 JPS=1,N
       IF(P(JPS,J).EQ.1)GO TO 950
  990  CONTINUE
```

```
 ISN          SOURCE STATEMENT

 122          JDIS=DD(KK,J)
 123          GO TO 960
 124   950    CONTINUE
 126   960    MDIS=DD(KK,K)
 127          KAKI=KAKI+1
 130          IKP(KAKI)=KK
 131          IPX(KAKI)=J
 132          NDIS(KAKI)=IDIS+JDIS-MDIS
 133          GO TO 922
 134    925   KAKI=KAKI+1
 135          NDIS(KAKI)=DD(I,K)
 136          IKP(KAKI)=100
 137          IPX(KAKI)=0
 140   922    CONTINUE
 142          CALL MIN(NDIS,KAKI,KIKI,MOA)
 143          P(I,KIKI)=1
 144          DO 930 IK=1,N
 145          IF(IK.EQ.I)GO TO 930
 150          P(IK,KIKI)=2
 151   930    CONTINUE
 153          DO 940 J=1,N
 154          IF(J.EQ.KIKI)GO TO 940
 157          P(I,J)=2
 160    940   CONTINUE
 162          LSP=IPX(KIKI)
 163          LTP=IXP(KIKI)
 164          IF(LTP.EQ.100)GO TO 920
 167          P(LTP,LSP)=1
 170          DO 970 II=1,N
 171           IF(LTP.EQ.II)GO TO 970
 174          P(II,LSP)=2
 175   970    CONTINUE
 177          DO 980 JJ=1,M
 200           IF(LSP.EQ.JJ)GO TO 980
 203          P(LTP,JJ)=2
 204    980   CONTINUE
 206   920    CONTINUE
 210          GO TO 995
 211   5000   KN=1000
 212   500    RETURN
 213          END
```

MEG250 IBMAP ASSEMBLY ASGNM

NO MESSAGES FOR ABOVE ASSEMBLY

```
 ISN       SOURCE STATEMENT

   0 $IBFTC ZERO                                                              PR
   1          SUBROUTINE ZERO(N)                                              PR
   2          INTEGER D,BD,ROW,COL,CA,RMIN,CMIN                               PR
   3          DIMENSION D(40,40),BD(50),ROW(40),COL(40)                       PR
   4          COMMON/ZLM/D                                                    PR
   5          DO 1 I=1,N                                                      PR
   6          DO 2 J=1,N                                                      PR
   7    2     BD(J)=D(I,J)                                                    PR
  11          CALL MIN(BD,N,KT,RMIN)                                          PR
  12          ROW(I)=RMIN                                                     PR
  13          DO 3 J=1,N                                                      PR
  14          CA=D(I,J)                                                       PR
  15          IF(CA.EQ.9999)GO TO 3                                           PR
  20          D(I,J)=D(I,J)-RMIN                                              PR
  21    3     CONTINUE                                                        PR
  23    1     CONTINUE                                                        PR
  25          DO 4 I=1,N                                                      PR
  26          DO 5 J=1,N                                                      PR
  27    5     BD(J)=D(J,I)                                                    PR
  31          CALL MIN(BD,N,KT,CMIN)                                          PR
  32          COL(I)=CMIN                                                     PR
  33          DO 6 J=1,N                                                      PR
  34          CA=D(J,I)                                                       PR
  35          IF(CA.EQ.9999)GO TO 6                                           PR
  40          D(J,I)=D(J,I)-CMIN                                              PR
  41    6     CONTINUE                                                        PR
  43    4     CONTINUE                                                        PR
  45          RETURN                                                          PR
  46          END                                                             PR
```

MCG250 IBMAP ASSEMBLY ZERO

NO MESSAGES FOR ABOVE ASSEMBLY

ISN SOURCE STATEMENT

```
$IBFTC ASSIGN
      SUBROUTINE ASSIGN(N,IPOK,ISOK,KOK)
      COMMON/JLM/P
      COMMON/ZLM/D
      INTEGER D,P
      DIMENSION D(40,40),P(40,40),LCOUNT(50),IPOK(40),ISOK(40)
      KOK=0
      LK=0
      INDEX=1
  990 NUM=0
      MUM=0
      DO 1 I=1,N
      DO 201 J=1,N
      LCOUNT(J)=0
  201 CONTINUE
      DO 11 K=1,N
      IF(D(I,K).EQ.0)GO TO 6
      GO TO 11
    6 IF(P(I,K).EQ.0)NUM=NUM+1
   11 CONTINUE
      IF(NUM.LE.INDEX)GO TO 3
      GO TO 1
    3 DO 2 J=1,N
      IF(D(I,J).EQ.0)GO TO 4
      LCOUNT(J)=10*N
      GO TO 2
    4 IF(P(I,J).EQ.0)GO TO 501
      LCOUNT(J)=10*N
      GO TO 2
  501 DO 102 IMC=1,N
      IF(D(IMC,J).NE.0)GO TO 102
      IF(P(IMC,J).NE.0)GO TO 102
      LCOUNT(J)=LCOUNT(J)+1
  102 CONTINUE
    2 CONTINUE
      CALL MIN(LCOUNT,N,J1,MINX)
      IF(MINX.EQ.10*N)GO TO 1
      P(I,J1)=1
      KOK=KOK+1
      IPOK(KOK)=I
      ISOK(KOK)=J1
      DO 5 II=1,N
      IF(II.EQ.I)GO TO 5
      P(II,J1)=2
    5 CONTINUE
      DO 8 JJ=1,N
      IF(JJ.EQ.J1)GO TO 8
      P(I,JJ)=2
    8 CONTINUE
    1 CONTINUE
      DO 301 J=1,N
      DO 302 I=1,N
      LCOUNT(I)=0
  302 CONTINUE
      DO 303 K=1,N
```

```
ISN        SOURCE STATEMENT

123        IF(D(K,J).EQ.0)GO TO 304
126        GO TO 303
127  304   IF(P(K,J).EQ.0)MUM=MUM+1
132  303   CONTINUE
134        IF(MUM.LE.INDEX)GO TO 305
137        GO TO 301
140  305   DO 306 I=1,N
141        IF(D(I,J).EQ.0)GO TO 307
144        LCOUNT(I)=10*N
145        GO TO 306
146  307   IF(P(I,J).EQ.0)GO TO 308
151        LCOUNT(I)=10*N
152        GO TO 306
153  308   DO 309 JMC=1,N
154        IF(D(I,JMC).NE.0)GO TO 309
157        IF(P(I,JMC).NE.0)GO TO 309
162        LCOUNT(I)=LCOUNT(I)+1
163  309   CONTINUE
165  306   CONTINUE
167        CALL MIN(LCOUNT,N,I1,NIMX)
170        IF(NIMX.EQ.10*N)GO TO 301
173        P(I1,J)=1
174        KOK=KOK+1
175        IPOK(KOK)=I1
176        ISOK(KOK)=J
177        DO 310 IK=1,N
200        IF(IK.EQ.I1)GO TO 310
203        P(IK,J)=2
204  310   CONTINUE
206        DO 311 JK=1,N
207        IF(JK.EQ.J)GO TO 311
212        P(I1,JK)=2
213  311   CONTINUE
215  301   CONTINUE
217        LP=0
220        DO 900 I=1,N
221        DO 900 J=1,N
222        IF(P(I,J).EQ.1)LP=LP+1
225  900   CONTINUE
230        IF(LP.EQ.LK)GO TO 901
233        LK=LP
234        GO TO 980
235  901   INDEX=INDEX+1
236  980   JOJ=0
237        DO 960 I=1,N
240        DO 960 J=1,N
241        IF(D(I,J).EQ.0)GO TO 970
244        GO TO 960
245  970   IF(P(I,J).EQ.0)JOJ=JOJ+1
250  960   CONTINUE
253        IF(JOJ.EQ.0)GO TO 999
256        GO TO 990
257  999   RETURN
260        END
```

MEG250 IBMAP ASSEMBLY ASSIGN

NO MESSAGES FOR ABOVE ASSEMBLY

ISN SOURCE STATEMENT

```
$IBFTC MINLIN
       SUBROUTINE MINLIN(UNMRKR,UR,MRRKC,UC,N)
       COMMON/ZLM/D
       COMMON/JLM/P
       INTEGER D,P,UNMRKR,UC,UR
       DIMENSION D(40,40),P(40,40),UNMRKR(40),MRKC(100),MRKR(100),MR(100)
       DIMENSION MC(100),JP(100),NINA(100),MRRKC(50)
       KM=0
       KL=0
       IR=0
       IC=0
C**** TO MARK ROWS HAVING NO ASSIGNED ZERO
       DO 1 I=1,N
       DO 2 J=1,N
       IF(P(I,J).EQ.1)GO TO 1
    2  CONTINUE
       KL=KL+1
       MR(KL)=I
    1  CONTINUE
       DO 101 I=1,KL
       IR=IR+1
       MRKR(IR)=MR(I)
  101  CONTINUE
C*'***TO MARK COLUMNS HAVING ZEROS IN ABOVE MARKED ROWS
       DO 3 I=1,N
       DO 4 J=1,KL
       IF(I.EQ.MR(J))GO TO 5
    4  CONTINUE
       GO TO 3
    5  DO 6 J=1,N
       IF(D(I,J).EQ.0)GO TO 7
       GO TO 6
    7  KM=KM+1
       MC(KM)=J
       IF(KM.EQ.1)GO TO 6
       KN=KM-1
       DO 21 II=1,KN
       IF(MC(KM).EQ.MC(II))GO TO 22
   21  CONTINUE
       GO TO 6
   22  KMS=KM
       MC(KMS)=0
       KM=KM-1
    6  CONTINUE
    3  CONTINUE
       DO 102 I=1,KM
       IC=IC+1
       MRKC(IC)=MC(I)
  102  CONTINUE
C***   TO MARK ROWS HAVING ASSIGNED ZEROS CORRESP TO ABOVE MARKED COL.
       CALL STEP1(N,MC,KM,JP,KTT)
       IF(KTT.EQ.0)GO TO 11
       DO 103 I=1,KTT
       IR=IR+1
       MRKR(IR)=JP(I)
```

```
 ISN        SOURCE STATEMENT

 110  103   CONTINUE                                                    PR
      C*****TO MARK COLUMNS HAVING ZEROS IN ABOVE MARKED LINES          PR
 112   200 CALL STEP2(N,JP,KTT,NINA,MODE)                               PR
 113        IF(MODE.EQ.0)GO TO 11                                       PR
 116        ID=IC                                                       PR
 117        IE=ID+1                                                     PR
 120        DO 104 J=1,MODE                                             PR
 121        IC=IC+1                                                     PR
 122        MRKC(IC)=NINA(J)                                            PR
 123  104   CONTINUE                                                    PR
 125        DO 105 J=1,ID                                               PR
 126        DO 106 JJ=IE,IC                                             PR
 127        IF(MRKC(J).EQ.MRKC(JJ))GO TO 150                            PR
 132        GO TO 106                                                   PR
 133  150   JT=JJ-ID                                                    PR
 134        NINA(JT)=0                                                  PR
 135        MRKC(JJ)=0                                                  PR
 136  106   CONTINUE                                                    PR
 140  105   CONTINUE                                                    PR
 142        CALL STEP1(N,NINA,MODE,JP,KTT)                              PR
 143        IF(KTT.EQ.0)GO TO 11                                        PR
 146        IT=IR                                                       PR
 147        IJ=IR+1                                                     PR
 150        DO 107 I=1,KTT                                              PR
 151        IR=IR+1                                                     PR
 152        MRKR(IR)=JP(I)                                              PR
 153  107   CONTINUE                                                    PR
 155        DO 109 II=1,KTT                                             PR
 156        DO 108 I=1,IT                                               PR
 157        IF(MRKR(I).EQ.JP(II))GO TO 160                              PR
 162        GO TO 108                                                   PR
 163  160   IM=IT+II                                                    PR
 164        MRKR(IM)=0                                                  PR
 165        JP(II)=0                                                    PR
 166  108   CONTINUE                                                    PR
 170  109   CONTINUE                                                    PR
 172        GO TO 200                                                   PR
 173   11   UR=0                                                        PR
 174        DO 301 I=1,N                                                PR
 175        DO 302 J=1,IR                                               PR
 176        IF(MRKR(J).EQ.I)GO TO 301                                   PR
 201  302   CONTINUE                                                    PR
 203        UR=UR+1                                                     PR
 204        UNMRKR(UR)=I                                                PR
 205  301   CONTINUE                                                    PR
 207        UC=0                                                        PR
 210        DO 303 J=1,N                                                PR
 211        DO 304 K=1,IC                                               PR
 212        IF(MRKC(K).EQ.J)GO TO 305                                   PR
 215  304   CONTINUE                                                    PR
 217        GO TO 303                                                   PR
 220  305   UC=UC+1                                                     PR
 221        MRRKC(UC)=J                                                 PR
 222  303   CONTINUE                                                    PR
 224        RETURN                                                      PR
```

```
MEG250                                                  FORTRAN SOURCE LIST MINLIN
    ISN        SOURCE STATEMENT

    225        END
MEG250                                                  IBMAP ASSEMBLY MINLIN

   NO MESSAGES FOR ABOVE ASSEMBLY
```

```
MEG250                                                    FORTRAN SOURCE LIST
    ISN         SOURCE STATEMENT

      0 $IBFTC STEP1
      1         SUBROUTINE STEP1(N,KR,KM,JP,KTT)
      2         COMMON/JLM/P
      3         COMMON/ZLM/D
      4         INTEGER D,P
      5         DIMENSION D(40,40),P(40,40),KR(100),JP(100)
      6         KT=0
      7         DO 9 J=1,N
     10         DO 10 IK=1,KM
     11    10   CONTINUE
     13         GO TO 9
     14    11   DO 12 I=1,N
     15         IF(P(I,J).EQ.1)GO TO 13
     20    12   CONTINUE
     22         GO TO 9
     23    13   KT=KT+1
     24         JP(KT)=I
     25     9   CONTINUE
     27         KTT=KT
     30         RETURN
     31         END
MEG250                                                  IBMAP ASSEMBLY STEP1


   NO MESSAGES FOR ABOVE ASSEMBLY
```

```
  ISN         SOURCE STATEMENT

    0 $IBFTC STEP2
    1         SUBROUTINE STEP2(N,JP,KTT,NINA,MODE)
    2         COMMON/JLM/P
    3         COMMON/ZLM/D
    4         DIMENSION D(40,40),P(40,40),JP(100),NINA(100)
    5         INTEGER D,P
    6         KD=0
    7         DO 14 I=1,N
   10         DO 15 KT=1,KTT
   11         IF(I.EQ.JP(KT))GO TO 16
   14    15   CONTINUE
   16         GO TO 14
   17    16   DO 18 J=1,N
   20         IF(D(I,J).EQ.0)GO TO 17
   23         GO TO 18
   24    17   IF(P(I,J).EQ.1)GO TO 18
   27         KD=KD+1
   30         NINA(KD)=J
   31         IF(KD.EQ.1)GO TO 18
   34         MLA=KD-1
   35         DO 30 II=1,MLA
   36         IF(NINA(KD).EQ.NINA(II))GO TO 31
   41    30   CONTINUE
   43         GO TO 18
   44    31   MLZP=KD
   45         NINA(MLZP)=0
   46         KD=KD-1
   47    18   CONTINUE
   51    14   CONTINUE
   53         MODE=KD
   54         RETURN
   55         END
```


NO MESSAGES FOR ABOVE ASSEMBLY

M_G250 FORTRAN SOURCE LIST

```
  ISN        SOURCE STATEMENT

    0 $IBFTC CRZERO
    1          SUBROUTINE CRZERO(N,UNMRKR,MRKC,UR,UC)
    2          COMMON/ZLM/D
    3          INTEGER D,UNMRKR,UC,UR
    4          DIMENSION D(40,40),UNMRKR(40),MRKC(50),IB(40,40),JB(50),JBJ(50)
    5.         KM=0
    6          DO 1 I=1,N
    7          DO 2 K=1,UR
   10          IF(UNMRKR(K).EQ.I)GO TO 1
   13    2     CONTINUE
   15          KM=KM+1
   16          KL=0
   17          DO 4 J=1,N
   20          DO 5 L=1,UC
   21          IF(MRKC(L).EQ.J)GO TO 4
   24    5     CONTINUE
   26          KL=KL+1
   27          IB(KM,KL)=D(I,J)
   30    4     CONTINUE
   32    1     CONTINUE
   34          DO 7 I=1,KM
   35          DO 8 J=1,KL
   36          JB(J)=IB(I,J)
   37    8     CONTINUE
   41          CALL MIN(JB,KL,KLN,MNL)
   42          JBJ(I)=MNL
   43    7     CONTINUE
   45          CALL MIN(JBJ,KM,KMN,MNM)
   46          DO 9 I=1,N
   47          DO 10 K=1,UR
   50          IF(UNMRKR(K).EQ.I)GO TO 11
   53   10     CONTINUE
   55          DO 12 J=1,N
   56          DO 13 L=1,UC
   57          IF(MRKC(L).EQ.J)GO TO 14
   62   13     CONTINUE
   64          IF(D(I,J).EQ.9999)GO TO 14
   67          D(I,J)=D(I,J)-MNM
   70          GO TO 12
   71   14     D(I,J)=D(I,J)
   72.  12     CONTINUE
   74          GO TO 9
   75   11     DO 15 J=1,N
   76          DO 21 L=1,UC
   77          IF(MRKC(L).EQ.J)GO TO 16
  102   21     CONTINUE
  104          D(I,J)=D(I,J)
  105          GO TO 15
  106   16     IF(D(I,J).EQ.9999)GO TO 15
  111          D(I,J)=D(I,J)+MNM
  112   15     CONTINUE
  114    9     CONTINUE
  116          RETURN
  117          END
```

MEG250 IBMAP ASSEMBLY CRZERO

NO MESSAGES FOR ABOVE ASSEMBLY

```
MEG250                                          FORTRAN SOURCE LIST
    ISN        SOURCE STATEMENT

      0 $IBFTC MIN
      1         SJBRJJTINE MIN(P,N,K,RMIN)
      2         INTEGER P,RMIN,SUM
      3         DIMENSION P(50)
      4         K=1
      5         SJM=P(1)
      6         IF(N.EQ.1)GO TO 2
     11         DJ 1 I=2,N
     12         IF(P(I).GE.SUM)GO TO 1
     15         K=I
     16         SJM=P(I)
     17    1    CJNTINUE
     21    2    RMIN=SUM
     22         RETURN
     23         END
MEG250                                          IBMAP ASSEMBLY MIN


   NO MESSAGES FJR ABOVE ASSEMBLY
MEG250                                          IBLDR -- JOB    000000

         ***  JBJECT PROGRAM IS BEING ENTERED INTO STORAGE AT  11 HRS.  22 MTS
```

```
      **************************************************************
      *                                                            *
      *  PROGRAM FOR ADJUSTING INITIAL ROUTES FOR CAPACITY *
      *  AND TIME CONSTRAINTS AND FOR REFINING THE ADJUSTED*
      *  ROUTES.PROGRAM ALSO CONSIDERS OMITTED CUSTOMERS.  *
      *                                                            *
      **************************************************************
C*****THIS PROGRAM ADJUSTS THE INITIAL ROUTS  TO BE WITHIN CAPACITY
C     AND TIME CONSTRAINT FOR LEAST INCREASE IN DISTANCE
C       AND THEN IMPROVES THE ROUTES BY REFINING HEURISTI3S
C*** THIS PROGRAM IS APPLICABLE TO BOTH TYPE OF VEHICLES
C*****N(I) DENOTES NO. OF NODES(EXCLUDING DEPOT IN SUBTOUR I
C*****TOTAL NO. OF NODES ARE NO. OF CUSTOMERS PLUS NO. OF ARTIFICIAL
C     DEPOTS WHICH ARE EQUAL TO NO. OF VEHICLES
C*****B(I,J) IS DISTANCE MATRIX  SHOWING DISTANCES BETWEEN ALL NODES
C*****D(.)  DENOTES DEMANDS OF ALL NODES.DEMANDS FOR ARTIFICIAL DEPOTS
C     ARE ZERO,THAT MEANS D(1),D(2),D(3) ETC. ARE ZERO.
C*****KRT(I,J) DENOTES JTH CUSTOMER IN ITH ROUTE    FOR EXAMPLE IN SUB-
C      TOUR 1-4-6-5-1, KRT(1,1)=4,KRT(1,2)=6 ETC.
C*****INITIAL TOURS OBTAINED IS,SAY,IN THE FORM-  1-4-6-5-3-10-11-1 THEN
C     SUBTOURS ARE  1-4-6-5-1 AND  3-10-11-3  PUT DATA IN THIS FORM
C     IS EQUAL TO 1 THEN ROUTE TRAVERSES FROM I TO J NODE
      DIMENSION N(10),MMX(40,40),KRT(10,30),D(40),B(40,40)
      DIMENSION JMX(40,40),OVC(10),UVC(10),KAPO(10),KEF(10),OOVC(10)
      DIMENSION NBKJ(40,40),JDB(40)
      DIMENSION KTOVC(10),C(10)
      DIMENSION IMEK(10)
      DIMENSION TCR(10),JRTD(10),JRT(10),JVT(10)
      INTEGER B,C,T,D,OVC,UVC,OOVC
      DIMENSION JST(40),JAV(10),CT(10),FX(10),JSET(10),JAL(10),CST(10)
      INTEGER TD
      COMMON/OPM/B,T,JMX,KRT
      COMMON/JLM/D,C
      COMMON/STS/NV,NN
 3081 FORMAT(2I3)
      READ 3081,(JAV(I),I=1,2)
      READ 3082,(CT(I),I=1,2)
 3082 FORMAT(2F6.1)
      READ 3083,(FX(I),I=1,2)
 3083 FORMAT(2F8.1)
 320  READ 101,NC
 101  FORMAT(I3)
      IF(NC.EQ.100)GO TO 7010
      NCC=NC+1
      DO 1580 I=1,NC
      JJ=I+1
      READ 1582,(NBKJ(I,II),II=JJ,NCC)
 1580  CONTINUE
 1582 FORMAT(20I4)
 902  FORMAT(1H1,5X,*SET POLICY*,I3/10X,*SET NUMBER*,I3/10X,*VEHICLES TO
     6 BE USED*,10I3)
 903  FORMAT(/10X,*NO. OF CUSTOMERS =*,I3)
      DO 103 I=1,NCC
      DO 103 J=1,NCC
      IF(J.GT.I)GO TO 103
      IF(J.EQ.I)GO TO 1588
      NBKJ(I,J)=NBKJ(J,I)
      GO TO 103
 1588 NBKJ(I,J)=9999
```

```
103   CONTINUE
      READ 1571,(JST(I),I=1,NCC)
1571  FORMAT(20I4)
      READ 340,(JDB(J),J=1,NCC)
340   FORMAT(20I4)
321   READ 7123,NPOLCY,MIPCY,CODE
7123  FORMAT(2I3,F6.1)
      IF(NPOLCY.EQ.100)GO TO 320
950   READ 910,NV
910   FORMAT(I3)
      IF(NV.EQ.500)GO TO 321
      READ 3088,(JSET(I),I=1,NV)
      NVV=NV+1
      NN=NV+NC
      DO 351 I=1,NV
      DO 351 J=1,NN
      IF(J.GT.NV)GO TO 356
      B(I,J)=9999
      GO TO 351
356   JJ=J-NV+1
      B(I,J)=NBKJ(1,JJ)
351   CONTINUE
      DO 352 I=NVV,NN
      II=I-NV+1
      DO 352 J=1,NN
      IF(J.GT.NV)GO TO 355
      B(I,J)=NBKJ(II,1)
      GO TO 352
355   JJ=J-NV+1
      B(I,J)=NBKJ(II,JJ)
352   CONTINUE
      DO 9100 I=1,NV
      OVC(I)=0
9100  UVC(I)=0
365   CONTINUE
367   DO 375 I=1,NN
      IF(I.LE.NV)GO TO 956
      II=I-NV+1
      D(I)=JDB(II)
      GO TO 375
956   D(I)=0
375   CONTINUE
377   CONTINUE
      RVP=BVP/100.
      DVP=RVP+1.0
      DO 7121 I=1,NN
      XLD=FLOAT(D(I))
7121  D(I)=IFIX(DVP*XLD)
      PRINT 7122,BVP,(D(I),I=1,NN)
7122  FORMAT(1H1,10X,*DEMANDS AT EACH NODE INCREASED BY*,F6.1,*PERCENT*,
     6/5X,20I5/5X,20I5)
380   READ 102,(N(I),I=1,NV)
102   FORMAT(10I4)
397   FORMAT(//5X,*INITIAL TOURS BY MULTIPLE TRAVELLING SALESMEN PROB*/)
1592  FORMAT(//20X,*DEMANDS ARE AS FOLLOWS*/5X,20I5/5X,20I5)
393   FORMAT(/8X,*SUBTOUR*,5X,*NO. OF NODES*,5X,*PATH SEQUENCE*/)
      DO 105 I=1,NV
      NPL=N(I)
      READ 106,(KRT(I,J),J=1,NPL)
```

```
 105   CONTINUE
 106   FORMAT(25I3)
 391   FORMAT(10X,I3,10X,I3,10X,25I3)
C*********  CODE=0.  FOR CONSTANT CAPACITY VEHICLE
C                CODE=1.    FOR  DIFFERENT SIZED VEHICLES
       JPOLCY=0
 8888  JPOLCY=JPOLCY+1
       IF(CODE.EQ.0.)GO TO 9227
       IF(JPOLCY.GT.3)GO TO 950
       PRINT 902,NPOLCY,MIPCY,(JSET(I),I=1,NV)
       PRINT 342,NV
 399   FORMAT(5X,*CONSIDERING VEHICLE-POLICY*,I3)
       GO TO 9228
 9227  IF(JPOLCY.GT.1)GO TO 950
       PRINT 902,NPOLCY,MIPCY,(JSET(I),I=1,NV)
       PRINT 342,NV
 9228  READ 100,T,(C(I),I=1,NV)
 100   FORMAT(15I5)
       READ 3085,(JAL(I),I=1,NV)
 3085  FORMAT(10I3)
 3088  FORMAT(10I3)
 8890  FORMAT(5X,*CAPACITIES OF VEHICLES*,10I5)
       IF(JPOLCY.GT.1)GO TO 901
        DO 200 I=1,NN
       DO 200 J=1,NN
       MMX(I,J)=0
 200   CONTINUE
       DO 201 I=1,NV
       MPL=N(I)
       JS=I
       DO 202 J=1,MPL
       KRTT=KRT(I,J)
       MMX(JS,KRTT)=1
       JS=KRTT
 202   CONTINUE
       MMX(JS,I)=1
 201   CONTINUE
 901   IPOLCY=0
       PRINT 399,JPOLCY
       PRINT 8890,(C(I),I=1,NV)
 900   DO 110 I=1,NN
       DO 110 J=1,NN
       JMX(I,J)=MMX(I,J)
 110   CONTINUE
       PRINT 8892
 8892  FORMAT(5X,33(* -*))
       IPOLCY=IPOLCY+1
       IF(IPOLCY.GT.3)GO TO 7000
C*****POLICY FOR CONSIDERING OVERLOADED ROUTES IN RANDOM ORDER
       PRINT 851,IPOLCY
 851   FORMAT(5X,*ROUTE-POLICY*,I3)
       MOTI=0
       CALL GUDIYA(JMX,KRT,N)
 1161  KEJ=0
       JEJ=0
 1001  CALL CAPCON(KRT,N,OVC,UVC)
       IF(IPOLCY.EQ.1)GO TO 1002
       GO TO 1100
```

```
      IF(OVC(I).EQ.0)GO TO 119
      JEJ=JEJ+1
      KAPO(JEJ)=I
      CALL SEARCH(N,OVC,UVC,KAPC,JEJ,KEP,KEJ)
      CALL CAPCON(KRT,N,OVC,UVC)
      IF(IPCLCY.EQ.1)GO TO 1002
      IF(IPCLCY.EQ.2)GO TO 1200
      IF(IPCLCY.EQ.3)GO TO 1300
119   CONTINUE
6000  CONTINUE                                                          PRO01321
      CALL DISCAL(TD)                                                   PRO01329
811   FORMAT(/5X,*TOTAL DISTANCE =*,I5)                                 PRO01331
      IF(KEJ.EQ.0)GO TO 800                                             PRO01332
      DO 960 I=1,KEJ                                                    PRO01351
      KAMO=KEP(I)                                                       PRO01352
      DO 961 II=1,NV                                                    PRO01353
      IF(UVC(II).EQ.0)GO TO 982                                         PRO01354
      IMEK(II)=D(KAMO)-UVC(II)                                          PRO01355
      GO TO 961                                                         PRO01356
982   IMEK(II)=5000                                                     PRO01357
961   CONTINUE                                                          PRO01358
      CALL MIN(IMEK,NV,NVH,IMK)                                         PRO01359
      IF(IMK.EQ.5000)GO TO 979                                          PRO01360
      KRZP=KRT(NVH,1)                                                   PRO01361
      JMX(NVH,KAMO)=1                                                   PRO01362
      JMX(NVH,KRZP)=0                                                   PRO01363
      JMX(KAMO,KRZP)=1                                                  PRO01364
      CALL GUDIYA(JMX,KRT,N)                                            PRO01365
      CALL CAPCON(KRT,N,OVC,UVC)                                        PRO01366
960   CONTINUE                                                          PRO01367
      MOTI=MOTI+1                                                       PRO01368
      IF(MOTI.EQ.21)GO TO 979                                           PRO01369
      GO TO 1161                                                        PRO01370
979   PRINT 973                                                         PRO01371
973   FORMAT(/10X,*OMMITED CUSTOMER REMAINS*)                           PRO01372
      GO TO 900                                                         PRO01373
800   CONTINUE                                                          PRO01379
850   FORMAT(//10X,*THERE IS NO CUSTOMERS LEFT UNASSIGNED*)
      CALL REFINE(N)                                                    PRO01381
      PRINT 1350                                                        PRO01382
1350  FORMAT(10X,*FINAL ROUTES OBTAINED ARE*)                           PRO01383
      PRINT 1360                                                        PRO01384
      PRINT 4618                                                        PRO01385
      NVN=NV-1                                                          PRO01386
      DO 1351 I=1,NV                                                    PRO01387
      NEP=N(I)                                                          PRO01388
      DO 1352 J=1,NEP                                                   PRO01389
      KRIM=KRT(I,J)                                                     PRO01390
      KRT(I,J)=KRIM-NVN                                                 PRO01392
1352  CONTINUE                                                          PRO01393
1351  CONTINUE                                                          PRO01394
      DO 3040 I=1,NV                                                    PRO01395
      LIZI=N(I)                                                         PRO0139
      NST=0                                                             PRO0139
      NID=0                                                             PRO0139
      JRTK=0                                                            PRO0139
      JSS=1                                                             PRO0140
      DO 3041 J=1,LIZI                                                  PRO0140
      KLIZ=KRT(I,J)                                                     PRO0140
```

```
      NST=NST+JST(KLIZ)
      NID=NID+JDB(KLIZ)
      JRTK=JRTK+NBKJ(JSS,KLIZ)
      JSS=KLIZ
 3041 CONTINUE
      JRTD(I)=JRTK+NBKJ(JSS,1)
      RTD=FLOAT(JRTD(I))/4.0
      UVC(I)=C(I)-NID
      IF(C(I).EQ.4000)GO TO 3059
      CST(I)=CT(2)*RTD
      TCR(I)=FX(2)+CST(I)
      JRT(I)=(15*JRTD(I)/JAV(2))+NST
      JVT(I)=JRT(I)+(2*JST(1))
      GO TO 3051
 3059 CST(I)=CT(1)*RTD
      TCR(I)=FX(1)+CST(I)
      JRT(I)=(15*JRTD(I)/JAV(1))+NST
      JVT(I)=JRT(I)+(2*JST(1))
 3051 II=1
      PRINT 1361,I,JAL(I),C(I),UVC(I),JRT(I),JVT(I),II,
     6(KRT(I,J),J=1,LIZI),II
 3040 CONTINUE
      TD=IFIX(RTD)
      CSTR=0.
      TCRT=0.
      DO 3029 I=1,NV
      CSTR=CSTR+CST(I)
      TCRT=TCRT+TCR(I)
 3029 CONTINUE
      PRINT 811,TD
      PRINT 3033,CSTR,TCRT
 3033 FORMAT(5X,*COST OF TRANSPORTATION=*,F9.2/5X,*TOTAL COST=*,F9.2)
      GO TO 900
 1100 IF(IPCLCY.EQ.2)GO TO 1200
      GO TO 1300
 1200 DO 1110 I=1,NV
      OOVC(I)=-OVC(I)
 1110 CONTINUE
 1360 FORMAT(5X,*ROUTE*,2X,*VEH. ALLOC.*,3X,*UVC*,3X,*RT*,3X,*VT*,5X,*RO
     6UTE PATH*)
 1361 FORMAT(6X,I2,2X,I4,4X,I4,3X,I4,1X,I4,1X,I4,5X,10I3)
 4618 FORMAT(12X,*NO.   CAP.*,7X,*(MTS.)*,1X,*(MTS)*)
 342  FORMAT(10X,*NO. OF VEHICLES =*,I3)
 1140 CALL MIN(OOVC,NV,NVO,MOVC)
      NOVC=-MOVC
      IF(NOVC.EQ.0) GO TO 6000
      JEJ=JEJ+1
      KAPO(JEJ)=NVO
 5000 CALL SEARCH(N,OVC,UVC,KAPO,JEJ,KEP,KEJ)
      CALL CAPCON(KRT,N,OVC,UVC)
      GO TO 1100
 1300 DO 1210 I=1,NV
      KTOVC(I)=OVC(I)
      IF(OVC(I).EQ.0)KTOVC(I)=999999
 1210 CONTINUE
      CALL MIN(KTOVC,NV,NVE,JOVC)
      IF(JOVC.EQ.999999)GO TO 6000
      IF(JOVC.EQ.9999)GO TO 6000
      JEJ=JEJ+1
```

```
      KAPO(JEJ)=NVE
      GO TO 5000
7000  GO TC 8888
7010  STOP
      END
```

```
    ISN        SOURCE STATEMENT

      0  $IBFTC SEARCH
      1         SUBROUTINE SEARCH(N,OVC,UVC,KAPO,JEJ,KEP,KEJ)
      2         INTEGER B,D,OVC,UVC,C,T,BB,A,OBJFN
      3         COMMON/STS/NV,NN
      4         COMMON/JLM/D,C
      5         COMMON/OPM/B,T,JMX,KRT
      6         DIMENSION KRT(10,30),N(10),OVC(10),UVC(10)
      7         DIMENSION KAPO(10),KEP(10),JMX(40,40)
     10         DIMENSION B(40,40),D(40),JEST(40),MINT(40),ISC(40),ICS(40)
     11         DIMENSION MINMUM(40),IV1(40),IV2(40),MINDIS(40),IW1(40),IW2(40)
     12         DIMENSION ITBEN(10,30),IB(10),A(10,40),BB(10),OBJFN(40),NX(40)
     13         DIMENSION NDIS(40),NXX(40),C(10)
     14         MGM=KAPO(JEJ)
     15         NVS=N(MGM)
     16         DO 2 J=1,NVS
     17         KL=0
     20         KY=KRT(MGM,J)
     21         DO 3 JJ=1,NN
     22         DO 1 JP=1,JEJ
     23         MGS=KAPO(JP)
     24         IF(D(KY).LE.UVC(MGS))GO TO 1
     27         IF(JJ.EQ.MGS)GO TO 3
     32         NVF=N(MGS)
     33         DO 101 JPS=1,NVF
     34         KZY=KRT(MGS,JPS)
     35         IF(JJ.EQ.KZY)GO TO 3
     40  101    CONTINUE
     42    1    CONTINUE
     44         IF(KEJ.EQ.0)GO TO 1520
     47         DO 1510 JAJC=1,KEJ
     50         MGZK=KEP(JAJC)
     51         IF(JJ.EQ.MGZK)GO TO 3
     54  1510   CONTINUE
     56  1520   IDIS=B(JJ,KY)
     57         KQ=0
     60         DO 5 JK=1,NN
     61         IF(JMX(JJ,JK).EQ.0)GO TO 5
     64         JDIS=B(KY,JK)
     65         MDIS=IDIS+JDIS
     66         KDIS=B(JJ,JK)
     67         KQ=KQ+1
     70         NDIS(KQ)=MDIS-KDIS
     71         JEST(KQ)=JK
     72    5    CONTINUE
     74         CALL MIN(NDIS,KQ,LSD,IMINA)
     75         KL=KL+1
     76         MINT(KL)=IMINA
     77         ISC(KL)=JJ
    100         ICS(KL)=JEST(LSD)
    101     3   CONTINUE
    103         IF(KL.EQ.0)GO TO 999
    106   995   CALL MIN(MINT,KL,LST,IMIN)
    107         IF(IMIN.EQ.9999)GO TO 999
    112         MINMUM(J)=IMIN
    113         IV1(J)=ISC(LST)
```

```
ISN       SOURCE STATEMENT

114            IV2(J)=ICS(LST)
115            IF(J.EQ.1)GO TO 998
120            JTA=J-1
121            DO 1082 JQA=1,JTA
122            IF(IV1(J).EQ.IV1(JQA))GO TO 997
125      1082  CONTINUE
127            GO TO 998
130       997  MINT(LST)=9999
131            GO TO 996
132       998  KEKE=0
133            IF(JMX(MGM,KY).EQ.0)GO TO 9
136            LKY=MGM
137            KEKE=KEKE+B(LKY,KY)
140            GO TO 8
141         9  DO 6 JJ=1,NVS
142            LKY=KRT(MGM,JJ)
143            IF(JMX(LKY,KY).EQ.0)GO TO 6
146            KEKE=KEKE+B(LKY,KY)
147            GO TO 8
150         6  CONTINUE
152         8  IF(JMX(KY,MGM).EQ.0)GO TO 10
155            MKY=MGM
156            KEKE=KEKE+B(KY,MKY)
157            GO TO 11
160        10  DO 12 JJ=1,NVS
161            MKY=KRT(MGM,JJ)
162            IF(JMX(KY,MKY).EQ.0)GO TO 12
165            KEKE=KEKE+B(KY,MKY)
166            GO TO 11
167        12  CONTINUE
171        11  KAKU=KEKE-B(LKY,MKY)
172            MINDIS(J)=IMIN-KAKU
173            IW1(J)=LKY
174            IW2(J)=MKY
175            GO TO 2
176       999  IMIN=9999
177            IV1(J)=0
200            IV2(J)=0
201            GO TO 998
202         2  CONTINUE
204            KINN=1
205            IF(JEJ.EQ.1)GO TO 20
210            JET=JEJ-1
211            DO 400 I=1,JET
212            KIN=0
213            MGST=KAPO(I)
214            NVFS=N(MGST)
215            DO 500 J=1,NVS
216            KZW=KRT(MGM,J)
217            DO 600 JJ=1,NVFS
220            IF(IV1(J).EQ.KRT(MGST,JJ))GO TO 700
223       600  CONTINUE
225            DO 750 JJ=1,NVFS
226            IF(IV2(J).EQ.KRT(MGST,JJ))GO TO 700
231       750  CONTINUE
```

```
ISN       SOURCE STATEMENT

          GO TO 450
     700  KIN=KIN+1
          ITEEN(KINN,J)=D(KZW)
     450  ITEEN(KINN,J)=0
     500  CONTINUE
          IF(KIN.EQ.0)GO TO 400
          IB(KINN)=UVC(MGST)
          KINN=KINN+1
     400  CONTINUE
      20  NCONS=KINN
          NVAR=NVS
          DO 2203 J=1,NVS
          KRTT=KRT(MGM,J)
          A(1,J)=D(KRTT)
    2203  CONTINUE
          BB(1)=OVC(MGM)
          IF(NCONS.EQ.1)GO TO 2500
          DO 2204 I=2,NCONS
          JIN=I-1
          DO 2206 J=1,NVAR
          BB(I)=IB(JIN)
          A(I,J)=ITEEN(JIN,J)
    2206  CONTINUE
    2204  CONTINUE
    2500  DO 2202 J=1,NVAR
          OBJFN(J)=MINDIS(J)
    2202  CONTINUE
          DO 840 I=1,NVS
     840  NK(I)=0
          CALL ZERO1(NVAR,NCONS,A,BB,OBJFN,NXX,IAB)
          DO 825 I=1,IAB
          NKT=NXX(I)
          NK(NXT)=1
     825  CONTINUE
          DO 6000 J=1,NVS
          KRTA=KRT(MGM,J)
          IF(NX(J).EQ.1)GO TO 6001
          GO TO 6000
    6001  IF(MINDIS(J).GT.9000)GO TO 6002
          IVV1=IV1(J)
          IVV2=IV2(J)
          JMX(IVV1,IVV2)=0
          JMX(IVV1,KRTA)=1
          JMX(KRTA,IVV2)=1
          GO TO 6003
    6002  KEJ=KEJ+1
          KEP(KEJ)=KRTA
          IVV1=0
           IVV2=0
    6003  IWW1=IW1(J)
          IWW2=IW2(J)
          IF(JMX(IWW1,KRTA).EQ.1)GO TO 6050
          IF(JMX(MGM,KRTA).EQ.1)GO TO 6069
          DO 6060 JJ=1,NVS
          KRAC=KRT(MGM,JJ)
```

```
 ISN        SOURCE STATEMENT

 346        IF(JMX(KRAC,KRTA).EQ.1)GO TO 6070
 351   6060 CONTINUE
 353   6069 KRAC=MGM
 354   6070 IWW1=KRAC
 355   6050 JMX(IWW1,KRTA)=0
 356        JMX(KRTA,IWW2)=0
 357        JMX(IWW1,IWW2)=1
 360   6000 CONTINUE
 362        CALL GUDIYA(JMX,KRT,N)
 363        RETURN
 364        END
```


NO MESSAGES FOR ABOVE ASSEMBLY

ISN SOURCE STATEMENT

```
$IBFTC CAPCON
      SUBROUTINE CAPCON(KRT,N,OVC,UVC)
      COMMON/STS/NV,NN
      COMMON/JLM/D,C
      INTEGER D,OVC,UVC,C,SUM
      DIMENSION KRT(10,30),D(40),SUM(10),OVC(10),UVC(10),N(10),C(10)
      DO 1 I=1,NV
      OVC(I)=0
      UVC(I)=0
      SUM(I)=0
      NVS=N(I)
      DO 2 J=1,NVS
      KRTT=KRT(I,J)
      SUM(I)=SUM(I)+D(KRTT)
    2 CONTINUE
      IF(SUM(I).LE.C(I))GO TO 3
      OVC(I)=SUM(I)-C(I)
      GO TO 1
    3 UVC(I)=C(I)-SUM(I)
    1 CONTINUE
      RETURN
      END
```

MEG250 IBMAP ASSEMBLY CAPCON

NO MESSAGES FOR ABOVE ASSEMBLY

```
     ISN        SOURCE STATEMENT

       0  $IBFTC MIN
       1        SUBROUTINE MIN(P,N,K,RMIN)
       2        INTEGER P,RMIN,SUM
       3        DIMENSION P(100)
       4        K=1
       5        SUM=P(1)
       6        IF(N.EQ.1)GO TO 2
      11        DO 1 I=2,N
      12        IF(P(I).GE.SUM)GO TO 1
      15        K=I
      16        SUM=P(I)
      17      1 CONTINUE
      21      2 RMIN=SUM
      22        RETURN
      23        END
```


NO MESSAGES FOR ABOVE ASSEMBLY

```
MEG250                                                FORTRAN SOURCE LIST
    ISN       SOURCE STATEMENT

      0  $IBFTC GUDIYA
      1         SUBROUTINE GUDIYA(JMX,KRT,N)
      2         COMMON/STS/NV,NN
      3         DIMENSION JMX(40,40),KRT(10,30),N(10)
      4         DO 1 I=1,NV
      5         K=0
      6         DO 2 J=1,NN
      7         IF(JMX(I,J).EQ.1)GO TO 3
     12     2   CONTINUE
     14     3   J2=J
     15     5   J1=J2
     16         K=K+1
     17         KRT(I,K)=J1
     20         DO 130 JJ=1,NN
     21         IF(JMX(J1,JJ).EQ.1)GO TO 4
     24   130   CONTINUE
     26     4   J2=JJ
     27         IF(J2.EQ.I)GO TO 6
     32         GO TO 5
     33     6   N(I)=K
     34     1   CONTINUE
     36         RETURN
     37         END
MEG250                                               IBMAP ASSEMBLY GUDIYA


   NO MESSAGES FOR ABOVE ASSEMBLY
```

```
  ISN        SOURCE STATEMENT

    0  $IBFTC ZERO1
    1          SUBROUTINE ZERO1(NN,MM,A,B,C,JOPT,IAB)
    2          INTEGER BMOST
    3          INTEGER Y,A,B,C,V,SORT
    4          INTEGER TEMP,ZMIN,CJSUM,ZOPT,SUMCR,APOS,YASUM,CFSUM
    5          DIMENSION JS(40),Y(10),JSUL(40),NS(40),ME(40),V(40),MF(40)
    6          DIMENSION JP(40),JR(10,40),NR(10),SORT(40),JOPT(40)
    7          DIMENSION INEQ(10),A(10,40),B(10),C(40),INVC(40)
   10          MSMAX=100
   11          MNMX=0
   12          INEQ(1)=1
   13          IF(MM.EQ.1)GO TO 5010
   16          DO 5000 I=2,MM
   17          INEQ(I)=2
   20   5000   CONTINUE
   22   5010   DO 43 I=1,40
   23          JP(I)=0
   24          JS(I)=0
   25          JSUL(I)=0
   26          SORT(I)=0.0
   27          DO 42 J=1,10
   30     42   JR(J,I)=0
   32     43   INVC(I)=0
       C***     READ PROBLEM, WRITE INITIAL TABLEAU
   34          CALL GERO(NN,MM,MNMX,INEQ,A,B,C,INVC,MSMAX)
   35          DO 49 I=1,MM
   36          NR(I) = 0
   37          DO 45 J=1,NN
   40          IF (A(I,J)) 45, 44, 44
   41     44   NR(I) = NR(I) +1
   42          IABC = NR(I)
   43          IF (A(I,J)) 4400,4400,4401
   44   4400   SORT(IABC) = 7777777
   45          GO TO 4402
   46   4401   SORT(IABC) =C(J)/A(I,J)
   47   4402   JR(I,IABC)=J
   50     45   CONTINUE
   52          IF (NR(I)-1) 49, 49, 46
   53     46   IAB=NR(I)-1
   54          DO 48 K=1,IAB
   55          KP1=K+1
   56          IABC=NR(I)
   57          DO 48 L=KP1,IABC
   60          IF (SORT(K)-SORT(L)) 48, 48, 47
   61     47   TEMP=SORT(K)
   62          SORT(K) = SORT(L)
   63          SORT(L)=TEMP
   64          ITEMP=JR(I,K)
   65          JR(I,K)=JR(I,L)
   66          JR(I,L)=ITEMP
   67     48   CONTINUE
   72     49   CONTINUE
   74          ZMIN=7777777
   75          MS=1
   76    300   BMOST = B(MM)
```

ISN SOURCE STATEMENT

```
      IMAX=1
      DO 304 I=1,MM
      IF (BMOST-B(I)) 302, 304, 304
302   BMOST=B(I)
      IMAX=I
304   CONTINUE
305   IABC=NR(IMAX)
      DO 325 I=1,IABC
      JS(I)=JR(IMAX,I)
      DO 319 K=1,MM
      Y(K)=-B(K)
      DO 318 L=1,I
      IAB=JS(L)
318   Y(K)=Y(K)+A(K,IAB)
      IF (Y(K)) 325,319,319
319   CONTINUE
      CJSUM=0.
      DO 323 J=1,I
      IAB=JS(J)
      CJSUM=CJSUM+C(IAB)
323   CONTINUE
      JRME=I
      GO TO 72
325   CONTINUE
      JRME=NR(IMAX)
      GO TO 100
52    MS=MS+1
C***   ITERATION COUNTER, EXIT IF MSMAX EXCEEDED
C
      IF (MS-MSMAX) 54, 54, 53
53    CONTINUE
      IF (JRME-39) 59, 59, 51
51    CONTINUE
59    K=1
      DO 354 J=1,NN
      IF (INVC(J)) 354,354,352
352   JP(K)=J
      K=K+1
354   CONTINUE
      IF (NN-39) 356,356,355
355   CONTINUE
356   GO TO 993
54    DO 57 IY=1,NN
      I=IY
      IF (JS(I)) 55, 58, 55
55    IF (NN-1) 57, 56, 57
56    I=I+1
      GO TO 58
57    CONTINUE
58    JRME=I-1
      CJSUM=0
      DO 64 I=1,JRME
      IF (JS(I)) 64, 64, 62
62    IAB=JS(I)
      CJSUM=CJSUM+C(IAB)
```

```
ISN        SOURCE STATEMENT

173   64   CONTINUE
175        DO 70 I=1,MM
176        Y(I)=-B(I)
177        DO 68 J=1,JRME
200        IF (JS(J)) 68, 68, 67
201   67   IAB=JS(J)
202        Y(I)=Y(I)+A(I,IAB)
203   68   CONTINUE
205   70   CONTINUE
207        DO 71 I=1,MM
210        IF (Y(I)) 100, 71,71
211   71   CONTINUE
213        IF (ZMIN-CJSUM) 73, 73, 72
214   72   ZMIN= CJSUM
215   73   CONTINUE
216   83   DO 76 J=1,JRME
217   76   JOPT(J)=JS(J)
221        IAB=JRME+1
222        DO 79 J=IAB,NN
223   79   JOPT(J)=0
225   74   IF (JSUL(JRME)) 75, 92, 92
226   75   DO 80 I=1,JRME
227        J=JRME+1-I
230        IF (JSUL(J)) 80, 85, 85
231   80   CONTINUE
     C***   PROBLEM COMPLETE, WRITE OUTPUT
     C
233        K=1
234        DO 84 J=1,NN
235        IF (INVC(J)) 84, 84, 82
236   82   JP(K)=J
237        K=K+1
240   84   CONTINUE
242   88   CONTINUE
243        DO 400 K=1,NN
244  400   JP(K)=0
246        DO 405 J=1,NN
247        IF (JOPT(J)) 401,405,402
250  401   IAB=-JOPT(J)
251        JP(IAB)=JOPT(J)
252        GO TO 405
253  402   IAB=JOPT(J)
254        JP(IAB)=JOPT(J)
255  405   CONTINUE
257        K=1
260        DO 412 J=1,NN
261        IF (JP(J)) 406,406,407
262  406   IF (INVC(J)) 412,412,409
263  407   IF (INVC(J)) 409,409,412
264  409   JOPT(K)=J
265        K=K+1
266  412   CONTINUE
270        IF (K-1-39) 413,413,414
271  413   IAD=K-1
272        GO TO 415
```

```
ISN        SOURCE STATEMENT

273   414  CONTINUE
274        IAB=K-1
275   415  ZOPT=0
276        IAB=K-1
277        DO 418 J=1,IAB
300        IABC = IABS(JOPT(J))
301        IF (INVC(IABC)) 417,417,416
302   416  ZOPT = ZOPT - C(IABC)
303        GO TO 418
304   417  ZOPT = ZOPT + C(IABC)
305   418  CONTINUE
307        IF (MNMX) 420,420,419
310   419  ZOPT=-ZOPT
311   420  CONTINUE
312        GO TO 993
313    85  JSUL(J)=-1
314        JS(J)=-JS(J)
315        IBC=J+1
316        DO 90 K=IBC,JRME
317        JS(K)=0
320    90  JSUL(K)=0
322        GO TO 52
323    92  JS(JRME)=-JS(JRME)
324        JSUL(JRME)=-1
325        GO TO 52
326   100  DO 102 J=1,NN
327   102  NS(J)=1
331        DO 106 J=1,JRME
332        IF (JS(J)) 103,106,104
333   103  JNEG=-JS(J)
334        NS(JNEG)=0
335        GO TO 106
336   104  IABC=JS(J)
337        NS(IABC)=0
340   106  CONTINUE
342        DO 120 I=1,NN
343        IF (NS(I)) 120,120,108
344   108  IF (C(I)-ZMIN+CJSUM) 120,110,110
345   110  NS(I)=0
346   120  CONTINUE
350   121  DO 122 I=1,NN
351   122  ME(I)=1
353        DO 140 I=1,MM
354        IF (Y(I)) 132,140,140
355   132  DO 135 J=1,NN
356        IF (A(I,J)) 135,135,134
357   134  ME(J)=0
360   135  CONTINUE
362   140  CONTINUE
364        DO 150 J=1,NN
365        IF (NS(J)) 150,150,144
366   144  IF (ME(J)) 150,150,146
367   146  NS(J)=0
370   150  CONTINUE
372        DO 160 J=1,NN
```

```
ISN        SOURCE STATEMENT

373        IF (NS(J)) 160,160,167
374   160  CONTINUE
376        GO TO 74
377   167  MARKF=0
400        DO 183 I=1,MM
401        IF (Y(I)) 169,183,183
402   169  SUMCR=0
403        APOS=0
404        IABC=NR(I)
405        DO 180 N=1,IABC
406        IAB=JR(I,N)
407        IF (NS(IAB)) 180,180,171
410   171  IAB=JR(I,N)
411        SUMCR=SUMCR+C(IAB)
412        APOS=APOS+A(I,IAB)
413        IF (SUMCR-ZMIN+CJSUM) 176,177,177
414   176  IF (APOS+Y(I)) 180,181,183
415   177  IF (APOS+Y(I)) 74, 74,183
416   180  CONTINUE
420        GO TO 74
421   181  IF (N-NR(I)) 173,182,182
422   173  IABC=NR(I)
423        DO 174 K=N,IABC
424        IAB=JR(I,K)
425        IF (NS(IAB)) 174,174,183
426   174  CONTINUE
430   182  MARKF=1
431   183  CONTINUE
433        IF (MARKF) 190,190,240
434   190  DO 210 J=1,NN
435        IF (NS(J)) 210,210,192
436   192  YASUM=0
437        DO 198 I=1,MM
440        IF (Y(I)+A(I,J)) 193,193,198
441   193  YASUM=YASUM+Y(I)+A(I,J)
442   198  CONTINUE
444        V(J)=YASUM
445   210  CONTINUE
447        DO 230 IY=1,NN
450         J=IY
451        IF (NS(J)) 230,230,218
452   218  IF (NN-J) 235,235,219
453   219  L=J+1
454        DO 226 K=L,NN
455        IF (NS(K)) 226,226,220
456   220  IF (V(J)-V(K)) 230,226,226
457   226  CONTINUE
461        DO 229 I=L,NN
462        IF (NS(I)) 229,229,225
463   225  IF (V(J)-V(I)) 227,227,229
464   227  IF (C(J)-C(I)) 229,229,228
465   228  J=I
466   229  CONTINUE
470        GO TO 235
471   230  CONTINUE
```

```
ISN         SOURCE STATEMENT

473  235    JS(JRME+1)=J
474         GO TO 52
475  240    DO 241 J=1,NN
476  241    MF(J)=0
500         DO 256 I=1,MM
501         IF (Y(I)) 243,256,256
502  243    APOS=0
503         DO 248 J=1,NN
504         IF (NS(J)) 248,248,245
505  245    IF (A(I,J)) 248,248,246
506  246    APOS=APOS+A(I,J)
507  248    CONTINUE
511         IF (APOS + Y(I)) 256, 249,256
512  249    DO 255 K=1,NN
513         IF (NS(K)) 255,255,250
514  250    IF (A(I,K)) 255,255,251
515  251    MF(K)=1
516  255    CONTINUE
520  256    CONTINUE
522         CFSUM=0
523         DO 262 I=1,NN
524         IF (MF(I)) 262,262,260
525  260    CFSUM=CFSUM+C(I)
526  262    CONTINUE
530         IF (CJSUM+CFSUM-ZMIN) 263,74, 74
531  263    IF (MS-1) 264,264,265
532  264    J=1
533         GO TO 267
534  265    J=JRME+1
535  267    DO 268 I=1,NN
536         IF (MF(I)) 268,268,266
537  266    JS(J)=I
540         J=J+1
541  268    CONTINUE
543         GO TO 52
544  993    RETURN
545         END
```

MEG250 IBMAP ASSEMBLY ZERO1

NO MESSAGES FOR ABOVE ASSEMBLY

```
MEG250                                                  FORTRAN SOURCE LIST
   ISN          SOURCE STATEMENT

     0  $IBFTC GERO
     1          SUBROUTINE GERO(NN,MM,MNMX,INEQ,A,B,C,INVC,MSMAX)
     2          DIMENSION INEQ(10),A(10,40),B(10),C(40),INVC(40)
     3          INTEGER A,B,C,SUMA
     4          MMM=MM
     5          IF (MNMX) 2,4,2
     6      2   DO 3 J=1,NN
     7      3   C(J)=-C(J)
    11      4   DO 25 J=1,MM
    12          IF (INEQ(J)-1) 25,25,5
    13      5   IF (INEQ(J)-2) 25,8,6
    14      6   IF (INEQ(J)-3) 25,15,15
    15      8   DO 10 K=1,NN
    16     10   A(J,K)=-A(J,K)
    20          B(J)=-B(J)
    21          GO TO 25
    22     15   MMM=MM+1
    23          DO 20 K=1,NN
    24     20   A(MMM,K)=-A(J,K)
    26          B(MMM)=-B(J)
    27     25   CONTINUE
    31          MM=MMM
    32          DO 30 J=1,NN
    33          IF (C(J)) 27,30,30
    34     27   C(J)=-C(J)
    35          INVC(J)=1
    36          DO 28 I=1,MM
    37     28   A(I,J)=-A(I,J)
    41     30   CONTINUE
    43          DO 35  I=1,MM
    44          SUMA=0
    45          DO 34 J=1,NN
    46          IF (INVC(J)) 34,34,32
    47     32   SUMA=SUMA+A(I,J)
    50     34   CONTINUE
    52          B(I)=B(I)+SUMA
    53     35   CONTINUE
    55          RETURN
    56          END
MEG250                                                 IBMAP ASSEMBLY GERO

  NO MESSAGES FOR ABOVE ASSEMBLY
```

MEG250 FORTRAN SOURCE LIST

```
    ISN        SOURCE STATEMENT

       0 $IBFTC DISCAL
       1        SUBROUTINE DISCAL(TD)
       2        COMMON/STS/NV,NN
       3        COMMON/OPM/B,T,JMX,KRT
       4        INTEGER TD,B,T
       5        DIMENSION JMX(40,40),B(40,40),KRT(10,30)
       6        TD=0
       7        DO 1 I=1,NN
      10        DO 2 J=1,NN
      11        IF(JMX(I,J).EQ.1)GO TO 3
      14      2 CONTINUE
      16      3 J1=J
      17        TD=TD+B(I,J1)
      20      1 CONTINUE
      22        RETURN
      23        END
```

MEG250 IBMAP ASSEMBLY DISCAL

NO MESSAGES FOR ABOVE ASSEMBLY

ISN SOURCE STATEMENT

```
$IBFTC REFINE
      SUBROUTINE REFINE(N)
C*****PROGRAM FOR 10 VEHICLES AND 30 CUSTOMERS
      DIMENSION N(10),D(40),B(40,40),KRT(10,30),MMX(40,40),JMX(40,40)
      DIMENSION NDIS(40),L1(40),L2(40),L(40),L3(40),OVC(10),UVC(10)
      DIMENSION MCUS(30),MNUS(30),MROUT(30,10),NW1(30),NW2(30),NKDIS(30)
      DIMENSION L4(40)
      DIMENSION NS1(30),NS2(40),NSS1(30),NSS2(30),NZDIS(30)
      DIMENSION C(10)
      DIMENSION NZOT(40)
      INTEGER C
      INTEGER TD,D,B,OVC,UVC,T
      COMMON/OPM/B,T,JMX,KRT
      COMMON/STS/NV,NN
      COMMON/JLM/D,C
      CALL DISCAL(TD)
      ITD=TD
      DO 10 I=1,NN
      DO 10 J=1,NN
      MMX(I,J)=JMX(I,J)
   10 CONTINUE
      DO 100 I=1,NV
      KSP=N(I)
      KSPP=KSP
      IF(KSP.EQ.1)GO TO 100
      DO 101 J=1,KSP
      IF(N(I).EQ.1)GO TO 100
      IF(KSPP.LT.J)GO TO 100
      KT=KRT(I,J)
      KQ=0
      DO 102 JJ=1,NN
      DO 103 KK=1,KSPP
      KTT=KRT(I,KK)
      IF(JJ.EQ.I)GO TO 102
      IF(JJ.EQ.KTT)GO TO 102
  103 CONTINUE
      IDIS=B(JJ,KT)
      DO 104 JK=1,NN
      IF(JMX(JJ,JK).EQ.1)GO TO 105
  104 CONTINUE
  105 JDIS=B(KT,JK)
      KDIS=B(JJ,JK)
      KQ=KQ+1
      NDIS(KQ)=IDIS+JDIS-KDIS
      L1(KQ)=JJ
      L2(KQ)=JK
      L(KQ)=KT
      DO 106 JJK=1,NN
      IF(JMX(JJK,KT).EQ.1)GO TO 107
      IF(JMX(KT,JJK).EQ.1)GO TO 108
      GO TO 106
  107 L3(KQ)=JJK
      GO TO 106
  108 L4(KQ)=JJK
  106 CONTINUE
```

```
 ISN         SOURCE STATEMENT

 113   102   CONTINUE
 115         CALL MIN(NDIS,KQ,KQQ,IMIN)
 116         NPS=L(KQQ)
 117         NP1=L1(KQQ)
 120         NP2=L2(KQQ)
 121         NP3=L3(KQQ)
 122         NP4=L4(KQQ)
 123         JMX(NP3,NPS)=0
 124         JMX(NPS,NP4)=0
 125         JMX(NP3,NP4)=1
 126         JMX(NP1,NPS)=1
 127         JMX(NPS,NP2)=1
 130         JMX(NP1,NP2)=0
 131         CALL SUDIYA(JMX,KRT,N)
 132         DO 109 II=1,NV
 133         NKST=N(II)
 134         DO 110 IK=1,NKST
 135         KMT=KRT(II,IK)
 136         IF(L1(KQQ).EQ.KMT)GO TO 111
 141         IF(L2(KQQ).EQ.KMT)GO TO 111
 144   110   CONTINUE
 146   109   CONTINUE
 150   111   NRT=II
       C*****NPS IS THE CUSTOMER ON ROUTE I WHICH IS TO BE SHIFTED TO SOME OTH
       C*****NP1 AND NP2 ARE NEW CONNECTIONS
       C*****NRT IS THE ROUTE TO WHICH CUSTOMER L(KQQ) IS SHIFTED
 151         KTZ=N(II)
 152         CALL CAPCON(KRT,N,OVC,UVC)
 153   7539 FORMAT(//5X,*SUBTOUR NO.*,5X,*OVC(.)*,5X,*UVC(.)*5X,*PATH SEQ.*/)
 154   7339 FORMAT(10X,I2,8X,I5,6X,I3,8X,30I3)
 155         IF(OVC(II).EQ.0)GO TO 4889
 160         KB=0
 161         DO 112 IR=1,KTZ
 162         KCT=KRT(II,IR)
 163         IF(D(KCT).GT.OVC(II))GO TO 113
 166         GO TO 112
 167   113   KB=KB+1
 170         MCUS(KB)=KCT
 171   112   CONTINUE
       C     FOR ACCOMODATING NPS IN IT) MCUS(.) IS ACTUAL CUSTOMER ON ROUTE
       C*****KB IS THE NO. OF CUSTOMERS PERMISSIBLE FOR SHIFTING FROM  ROUTE
       C**   NOW CHECK WHICH CUSTOMERS OUT OF ABOVE MCUS(.) CAN BE SHIFTED TO
       C      OTHER ROUTES AND FOR EACH OF THESE CUSTOMERS, NOTE THE POSSIBLE
 173         KBB=0
 174         IF(KB.EQ.0)GO TO 5879
 177         KBR=0
 200         DO 114 IU=1,KB
 201         MCUST=MCUS(IU)
 202         KBB=KBB+1
 203         KBL=0
 204         DO 115 IT=1,NV
 205         IF(IT.EQ.II)GO TO 115
 210         IF(UVC(IT).GE.D(MCUST))GO TO 116
 213         GO TO 115
 214   116   KBL=KBL+1
```

```
ISN        SOURCE STATEMENT

215              IF(KBL.GE.2)GO TO 1139
220              KBR=KBR+1
221              NZOT(KBR)=KBL
222              MNUS(KBR)=MCUST
223        1139  MROUT(KBR,KBL)=IT
224        115   CONTINUE
226        114   CONTINUE
           C*****  KBR IS THE NO. OF CUSTOMERS ON ROUTE II WHICH CAN BE SHIFTED TO
           C        OTHER ROUTES AND DELETION OF WHICH CAN ACCOMODATE NPS FROM I ON
230              IF(KBR.EQ.0)GO TO 5879
           C***    ABOVE STATEMENT MEANS NO CUSTOMER CAN BE DELETED FROM ROUTE II
           C       SO CONSIDER NEXT CUSTOMER ON ROUTE I.
233              DO 117 IZ=1,KBR
234              MSC=MNUS(IZ)
235              DO 118 IWK=1,NN
236              IF(JMX(IWK,MSC).EQ.1)GO TO 119
241              GO TO 118
242        119   NW1(IZ)=IWK
243        118   CONTINUE
245              DO 1118 IWKQ=1,NN
246              IF(JMX(MSC,IWKQ).EQ.1)GO TO 120
251               GO TO 1118
252        120   NW2(IZ)=IWKQ
253        1118  CONTINUE
255        117   CONTINUE
257              DO 121 IZ=1,KBR
260              MSC=MNUS(IZ)
261              NPLM=NZOT(IZ)
262              KCONT=0
263              DO 122 IZK=1,NN
264              DO 123 IUK=1,NPLM
265              KROUT=MROUT(IZ,IUK)
266              NUW=N(KROUT)
267              DO 124 JUK=1,NUW
270              KARR=KRT(KROUT,JUK)
271              IF(IZK.EQ.KARR)GO TO 125
274              GO TO 124
275        125   MKDIS=B(KARR,MSC)
276              DO 126 IWKK=1,NN
277              IF(JMX(KARR,IWKK).EQ.1)GO TO 127
302        126   CONTINUE
304        127   JKDIS=B(MSC,IWKK)
305              KKDIS=B(KARR,IWKK)
306              KCONT=KCONT+1
307              NKDIS(KCONT)=MKDIS+JKDIS-KKDIS
310              NS1(KCONT)=KARR
311              NS2(KCONT)=IWKK
312        124   CONTINUE
314        123   CONTINUE
316        122   CONTINUE
320              CALL MIN(NKDIS,KCONT,KCENT,JIN)
321              NSS1(IZ)=NS1(KCENT)
322              NSS2(IZ)=NS2(KCENT)
323              NATO=NW1(IZ)
324              MATO=NW2(IZ)
```

```
   ISN        SOURCE STATEMENT

   325             MOPIS=B(NATO,MSC)
   326             NOPIS=B(MSC,MATO)
   327             KOPIS=B(NATO,MATO)
   330             MZDIS=MOPIS+NOPIS-KCPIS
   331             NZDIS(IZ)=JIN-MZDIS
   332    121      CONTINUE
   334             CALL MIN(NZDIS,KBR,KBRR,JINE)
   335             NY=MNUS(KBRR)
   336             NY1=NW1(KBRR)
   337             NY2=NW2(KBRR)
   340             NY3=NSS1(KBRR)
   341             NY4=NSS2(KBRR)
   342             JMX(NY1,NY2)=1
   343             JMX(NY1,NY)=0
   344             JMX(NY,NY2)=0
   345             JMX(NY3,NY)=1
   346             JMX(NY,NY4)=1
   347             JMX(NY3,NY4)=0
   350             CALL GUDIYA(JMX,KRT,N)
   351             CALL CAPCON(KRT,N,OVC,UVC)
   352    4889     CONTINUE
   353             CALL DISCAL(TD)
   354             NTD=TD
   355             IF(NTD.LT.ITD)GO TO 160
   360    5879     DO 140 IES=1,NN
   361             DO 140 JES=1,NN
   362             JMX(IES,JES)=MMX(IES,JES)
   363    140      CONTINUE
   366             GO TO 150
   367    160      DO 200 IFZ=1,NN
   370             DO 200 IEW=1,NN
   371             MMX(IFZ,IEW)=JMX(IFZ,IEW)
   372    200      CONTINUE
   375             ITD=NTD
   376    150      CALL GUDIYA(JMX,KRT,N)
   377             CALL CAPCON(KRT,N,OVC,UVC)
   400             KSPP=N(I)
   401    101      CONTINUE
   403    100      CONTINUE
   405             CALL GUDIYA(JMX,KRT,N)
   406             CALL CAPCON(KRT,N,OVC,UVC)
   407             RETURN
   410             END
MEG250                                              IBMAP ASSEMBLY REFINE

   NO MESSAGES FOR ABOVE ASSEMBLY
MEG250                                              IBLDR -- JOB    000000

          ***  OBJECT PROGRAM IS BEING ENTERED INTO STORAGE AT  11 HRS.  27 MI
```

```
      ******************************************************************
      *                                                                *
      *   SHORTEST PATH ALGORITHM                                      *
      *                                                                *
      ******************************************************************

      DIMENSION D(30,30)
      INTEGER D
      READ 100,N
100   FORMAT(I3)
      NIN=N-1
      DO 101 I=1,NIN
      JJ=I+1
      READ 102,(D(I,II),II=JJ,N)
101   CONTINUE
102   FORMAT(20I4)
      DO 200 I=1,N
      DO 200 J=1,N
      IF(J.GT.I)GO TO 200
      IF(J.EQ.I)GO TO 201
      D(I,J)=D(J,I)
      GO TO 200
201   D(I,J)=0
200   CONTINUE
      DO 5010 ITR=1,6
      PRINT 400,(I,I=1,N)
400   FORMAT(1H1,////25X,*DISTANCE MATRIX*//11X,30I4/)
      DO 103 I=1,N
      PRINT 104,I,(D(I,J),J=1,N)
103   CONTINUE
104   FORMAT(5X,I2,5X,30I4)
      PRINT 500
500   FORMAT(//10X,*NOTE- D(I,J) MEANS SHORTEST DIRECT LINK BETWEEN
     6I AND J NODE*,/10X,*      D(I,J)=999 MEANS NO DIRECT LINK BETWEEN
     6I AND J NODE*)
5010  CONTINUE
      CALL FLOYD(D,N)
      DO 6010 JTR=1,6
      PRINT 300,(I,I=1,N)
300   FORMAT(1H1,////25X,*SHORTEST DISTANCE MATRIX*//11X,30I4/)
      DO 105 I=1,N
       PRINT 106,I,(D(I,J),J=1,N)
105   CONTINUE
106   FORMAT(5X,I2,5X,30I4)
6010  CONTINUE
      PRINT 700
700   FORMAT(1H1,//*FINISH*)
      STOP
      END
£IBFTC FLOYD
      SUBROUTINE FLOYD(D,N)
      INTEGER D,S
      DIMENSION D(30,30)
      DIMENSION IP(30,30)
      DO 15 I=1,N
       DO 15 J=1,N
      IP(I,J)=0
15    CONTINUE
```

```
      DO 12 K=1,N
      DO 12 I=1,N
      IF(D(I,K).EQ.999)GO TO 12
      DO 11 J=1,N
      IF(D(K,J).EQ.999)GO TO 11
      S=D(I,K)+D(K,J)
      IF(S.LT.D(I,J))GO TO 50
      GO TO 11
   50 D(I,J)=S
      IP(I,J)=K
   11 CONTINUE
   12  CONTINUE
      RETURN
      END
```

```
      ***************************************************************
      *                                                             *
      *      CLARKE AND WRIGHTS METHOD                              *
      *                                                             *
      ***************************************************************


C*******    THIS PROGRAM IS FOR 34 CUSTOMERS AND 1 DEPOT
      DIMENSION D(35),B(35,35),C(15),KOP(35),JMX(35,35),IB(35),JB(35)
      DIMENSION DROUT(15),KB(35),KIP(35),JOD(35)
      DIMENSION KRT(15,35),N(15)
      INTEGER B,D,DROUT,C
      COMMON/SITA/NT,NN,JMX,B
C***** NC IS NO. OF CUSTOMERS AND NT IS TOTAL NO . OF VEHICLES AVAILABLE
C     WHICH MIGHT BE LESS OR MORE THAN ACTUAL VEHICLES NEEDED
      READ 1,NC,NT
    1 FORMAT(2I3)
      IF(NC.EQ.100)GO TO 9000
      NN=NC+1
C*******FOR DEPOT DEMAND IS ZERO, D(1)=0
      DO 3 I=1,NC
      JJ=I+1
      READ 4,(B(I,II),II=JJ,NN)
    3 CONTINUE
    4 FORMAT(20I4)
      DO 1020 I=1,NN
      DO 1020 J=1,NN
      IF(J.GT.I)GO TO 1020
      IF(J.EQ.I)GO TO 1021
      B(I,J)=B(J,I)
      GO TO 1020
 1021 B(I,J)=999
 1020 CONTINUE
      IPOL=0
 8049 IPOL=IPOL+1
      IF(IPOL.GT.4)GO TO 9000
      PRINT 250
  250 FORMAT(1H1,/10X,*ROUTES OBTAINED BY CLARKE AND WRIGHT METHOD*)
      READ 2,(D(J),J=1,NN)
    2 FORMAT(20I4)
      JPOLCY=0
    8 JPOLCY=JPOLCY+1
      PRINT 5050
 5050 FORMAT(5X,33(* -*))
      IF(JPOLCY.GT.3)GO TO 8049
 1060 READ 1001,(C(I),I=1,NT)
 1001 FORMAT(10I6)
      PRINT 1550,JPOLCY
 1550 FORMAT(/10X,*CONSIDERING VEHICLE-POLICY*,I3)
      PRINT 2005,(C(I),I=1,NT)
 2005 FORMAT(10X,*CAPACITIES*,10I5/)
      K=0
      KROUT=0
      DO 101 JJK=1,NN
  101  KOP(JJK)=0
      DO 9 I=1,NN
      DO 9 J=1,NN
      JMX(I,J)=0
    9 CONTINUE
```

```
700   IB(1)=9999
      DO 10 I=2,NN
      DO 102 JJK=1,NN
      IF(KOP(JJK).EQ.I)GO TO 103
102   CONTINUE
      GO TO 104
103   IB(I)=9999
      GO TO 10
104   IB(I)=B(1,I)
 10   CONTINUE
      CALL MIN(IB,NN,NK,IBB)
      IF(IBB.EQ.9999)GO TO 4000
C*****  NK IS THE NEAREST CUSTOMER TO DEPOT AMONG UNASSIGNED CUSTOMERS
      KROUT=KROUT+1
      IF(KROUT.GT.NT)GO TO 4000
      JMX(1,NK)=1
      DROUT(KROUT)=D(NK)
500   K=K+1
      KOP(K)=NK
C****  JB(I) IS SAVINGS OBTAINED BY CONNECTING NK TO I
      JB(1)=-9999
      JIBB=B(1,NK)
      DO 11 I=2,NN
      DO 12 KK=1,K
      IF(I.EQ.KOP(KK))GO TO 13
 12   CONTINUE
      JB(I)=JIBB+B(I,1)-B(NK,I)
      GO TO 11
 13   JB(I)=-9999
 11   CONTINUE
      DO 107 I=1,NN
107   KB(I)=-JB(I)
      MKP=0
 16   CALL MIN(KB,NN,NKK,KBB)
      IF(KBB.EQ.9999)GO TO 23
      IF(KBB.EQ.999)GO TO 15
      MKP=MKP+1
      KIP(MKP)=NKK
      KB(NKK)=999
      GO TO 16
 15   CONTINUE
      DO 17 I=1,MKP
      NKT=KIP(I)
      JOD(I)=JB(NKT)
 17   CONTINUE
      DO 20 I=1,MKP
      NKT=KIP(I)
      DROUT(KROUT)=DROUT(KROUT)+D(NKT)
       IF(DROUT(KROUT).GT.C(KROUT))GO TO 21
      GO TO 22
 21   DROUT(KROUT)=DROUT(KROUT)-D(NKT)
 20   CONTINUE
      GO TO 23
 22   JMX(NK,NKT)=1
      NK=NKT
      GO TO 500
 23   JMX(NK,1)=1
      GO TO 700
4000  CONTINUE
```

```
0   IB(1)=9999
    DO 10 I=2,NN
    DO 102 JJK=1,NN
    IF(KOP(JJK).EQ.I)GO TO 103
2   CONTINUE
    GO TO 104
3   IB(I)=9999
    GO TO 10
4   IB(I)=B(1,I)
0   CONTINUE
    CALL MIN(IB,NN,NK,IBB)
    IF(IBB.EQ.9999)GO TO 4000
***  NK IS THE NEAREST CUSTOMER TO DEPOT AMONG UNASSIGNED CUSTOMERS
    KROUT=KROUT+1
    IF(KROUT.GT.NT)GO TO 4050
    JMX(1,NK)=1
    DROUT(KROUT)=D(NK)
0   K=K+1
    KOP(K)=NK
**  JB(I) IS SAVINGS OBTAINED BY CONNECTING NK TO I
    JB(1)=-9999
    JIBB=B(1,NK)
    DO 11 I=2,NN
    DO 12 KK=1,K
    IF(I.EQ.KOP(KK))GO TO 13
2   CONTINUE
    JB(I)=JIBB+B(I,1)-B(NK,I)
    GO TO 11
3   JB(I)=-9999
1   CONTINUE
    DO 107 I=1,NN
7   KB(I)=-JB(I)
    MKP=0
6   CALL MIN(KB,NN,NKK,KBB)
    IF(KBB.EQ.9999)GO TO 23
    IF(KBB.EQ.999)GO TO 15
    MKP=MKP+1
    KIP(MKP)=NKK
    KB(NKK)=999
    GO TO 16
5   CONTINUE
    DO 17 I=1,MKP
    NKT=KIP(I)
    JOD(I)=JB(NKT)
7   CONTINUE
    DO 20 I=1,MKP
    NKT=KIP(I)
    DROUT(KROUT)=DROUT(KROUT)+D(NKT)
     IF(DROUT(KROUT).GT.C(KROUT))GO TO 21
    GO TO 22
1   DROUT(KROUT)=DROUT(KROUT)-D(NKT)
0   CONTINUE
    GO TO 23
2   JMX(NK,NKT)=1
    NK=NKT
    GO TO 500
23  JMX(NK,1)=1
    GO TO 700
000 CONTINUE
```

```
      KN=0
      CALL BABBO(KRT,N,IDIS,KN)
      II=1
      DO 410 I=1,KN
      NPS=N(I)
      PRINT 420,I,II,(KRT(I,J),J=1,NPS),II
20    FORMAT(10X,*SUBTOUR*,I3,5X,30I3)
10    CONTINUE
      PRINT 260,IDIS
60    FORMAT(/10X,*TOTAL DISTANCE =*,I4)
      PRINT 320,KN
20    FORMAT(10X,*REQD. VEHICLES =*,I4)
      GO TO 8
050   PRINT 290
90    FORMAT(//20X,*REQD. VEHICLES(ROUTES) ARE MORE THAN AVAILABLE VEH*)
      GO TO 8
000   STOP
      END
FTC BABBO
      SUBROUTINE BABBO(KRT,N,IDIS,KN)
      COMMON/SITA/NT,NN,JMX,B
      INTEGER B
       DIMENSION B(35,35),JMX(35,35),N(15),KRT(15,35)
      I=0
      IDIS=0
      DO 2 J=1,NN
      K=0
      IF(JMX(1,J).EQ.1)GO TO 3
      GO TO 2
      J2=J
      I=I+1
      IDIS=IDIS+B(1,J2)
 5    J1=J2
      K=K+1
      KRT(I,K)=J1
      DO 130 JJ=1,NN
      IF(JMX(J1,JJ).EQ.1)GO TO 4
0     CONTINUE
 4    J2=JJ
      IDIS=IDIS+B(J1,J2)
      IF(J2.EQ.1)GO TO 6
      GO TO 5
 6    N(I)=K
 2    CONTINUE
      KN=I
      RETURN
      END
FTC MIN
      SUBROUTINE MIN(P,N,K,RMIN)
      INTEGER P,RMIN,SUM
      DIMENSION P(50)
      K=1
      SUM=P(1)
      IF(N.EQ.1)GO TO 2
      DO 1 I=2,N
      IF(P(I).GE.SUM)GO TO 1
      K=I
      SUM=P(I)
 1     CONTINUE
```

ME-1974-M-MOD-DET